A Comparative Study of Deep Learning Methods for Infant Cry Detection

Qingyu Liu[1], Li Li[1], Yifan Zhou[1], Dongxing Xu[2], and Yanhua Long[1,2⋆]

[1] Shanghai Engineering Research Center of Intelligent Education and Bigdata, Shanghai Normal University, Shanghai, China
[2] Unisound AI Technology Co., Ltd., Beijing, China
1000500956@smail.shnu.edu.cn,yanhua@shnu.edu.cn

Abstract. In the initial months after birth, infants rely solely on crying as their primary means of communication with the external world. However, for parents of newborns, deciphering the meaning behind an infant's cries can be a challenging task. In this study, we extract spectrograms of each category of infant cry as feature inputs and utilize CNN, CRNN, AST, and the ensemble model to classify five categories of infant cries using the Baby Chillanto Database. Three distinct tasks with different training and testing conditions are designed for evaluation of model robustness. Experimental results show that the ensemble model ultimately achieves the best performance, with accuracies of 94.05% and 61.10% on the clean and noisy test sets respectively, thereby presenting compelling evidence for the complementary information between the three individual deep learning models with heterogeneous network architectures. This research offers a practical solution to aid parents in effectively understanding their baby's condition and needs.

Keywords: Infant cry detection · CNN · CRNN · AST · Ensemble modeling.

1 Introduction

For newborn babies, crying is their only way of communicating with the outside world. However, many parents encounter challenges in accurately interpreting their baby's emotions or needs. Furthermore, during the initial months after birth, accurate diagnosis of conditions like asphyxia and deafness requires lengthy testing procedures, which can lead to irreversible brain damage in some infants. To address these concerns, infant cry detection technology has emerged as a potential solution. By conducting research on infant cry detection, advancements can be achieved in domains including infant care, monitoring newborn health status, and screening for neonatal diseases.

Research on infant cry detection was initiated by Wasz-Hockert et al in the 1960s [1]. In the literature, many previous studies focused on using conventional machine learning techniques. For example, in [2], the incremental support

⋆ Yanhua Long is the corresponding author, she is also with the Key Innovation Group of Digital Humanities Resource and Research, Shanghai Normal University.

vector machine (SVM) was proposed for infant cry recognition; Authors in [3] introduced a KNN classifier that utilizes Pitch, Short-time energy, Harmonicity Factor, and Harmonic-to-Average Power as features for infant cry signal classification; Works in [4] attempted to apply Gaussian Mixture Models (GMM) to classify distinct categories of infant vocalizations, while [5] used a type-2 fuzzy classifier for classifying hunger and pain infant cries. In recent years, with the increasing popularity of deep learning, numerous studies have endeavored to employ neural networks for infant cry detection. Such as, in [6], they utilized DNN and CNN to effectively classify infant cry into non-asphyxia and asphyxia categories; Works in [7] applied the large-scale self-supervised pre-training to assist the cry-based detection of neurological injury as well as the identification of cry triggers. In summary, the transition from conventional machine learning to deep learning has been evident in the study of infant cry detection. However, most of these previous works only focused on the clean condition of infant cry detection, investigation of the noise robustness and cross-domain generalization ability of DNN-based models is extremely limited.

In this study, we aim to explore the cross-domain noise robustness and behavior of individual DNN-based models with heterogeneous network architectures for infant cry detection. The main contributions are as follows: 1) Three different types of DNN-based models, the CNN, CRNN, and AST models are investigated for the infant cry detection; 2) The ensemble learning is proposed to combine three individual models to exploit they complementary information; 3) We design three distinct tasks, namely Clean-Clean, Clean-Noise and CNoise-CNoise to evaluate the methods' cross-domain and noise robustness on both clean and noisy conditions. Experimental results show that across three distinct tasks, the three heterogeneous DNN models show different advantages under different infant cry detection conditions. The complementary information between individual models can be well captured by the ensemble model.

2 Methods

2.1 Overall Framework

Ensemble learning is a machine learning technique that combines multiple models to enhance the accuracy and stability of predictions, finding extensive applications in diverse domains such as computer vision [8] and natural language processing [9].

In this study, we first design three individual deep neural networks with heterogeneous structures to examine their effectiveness for five categories of infant cry detection, then we apply ensemble learning to further enhance the infant cry detection performance. The overall framework is illustrated in Fig. 1. Specifically, the three individual models are a simple CNN (Convolutional Neural Network), a CRNN (convolutional neural network plus a recurrent neural network), and an AST (Audio Spectrogram Transformer) model. We employ soft voting to construct the ensemble model by calculating a weighted average of posterior

probabilities on five categories predicted by the CNN, CRNN, and AST models. The final prediction outcome corresponds to the category with the highest weighted average probability. Below are the details:

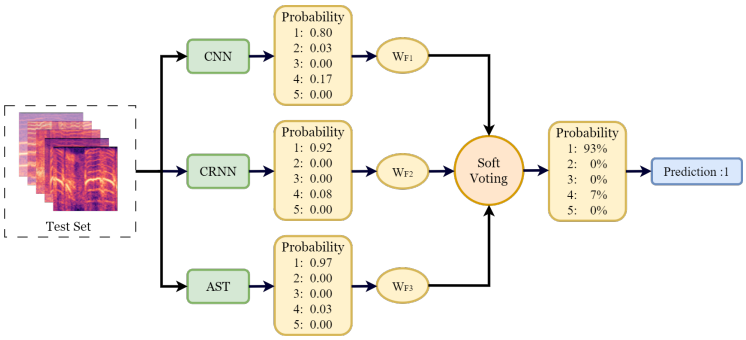

Fig. 1. Structure of the ensemble model

$$P_j = \frac{\sum_{i=1}^{N} P_{ij} W_{Fi}}{\sum_{i=1}^{N} W_{Fi}} \tag{1}$$

where P_{ij} represents the prediction probability of the i model for class j, and W_{Fi} represents the weight assigned to model i. N represents the number of individual deep learning models we utilize.

The details of the specific structures of designed CNN, CRNN, and AST models for infant cry detection are presented in the following sections.

2.2 CNN

CNN (Convolutional Neural Network) exhibits superiority in infant cry detection tasks due to its ability to automatically learn hierarchical features from raw audio data. By employing convolutional layers, CNN captures local patterns and gradually assembles them into higher-level representations, enabling effective discrimination between different cry patterns. This hierarchical feature learning makes CNN well-suited for recognizing complex patterns in infant cries, contributing to its superior performance in the task of infant cry detection.

In previous works, CNN has been extensively studied for the infant cry detection purpose and has consistently showcased commendable performance. In [10], CNN achieved an impressive recognition rate of 89% on the Dunstan Baby Language Database. Additionally, works in [11] compared the CNN with ANN, and LSTM on an infant cry dataset from Far Eastern Memorial Hospital, and they found that the CNN consistently outperformed other models in four-class, three-class, and two-class infant cry classification tasks.

In this paper, we also design a CNN network to see its effectiveness of our cross-domain infant cry detection, the detail structure is shown in Fig. 2. The architecture is composed of two main parts: the convolutional component and the linear component.

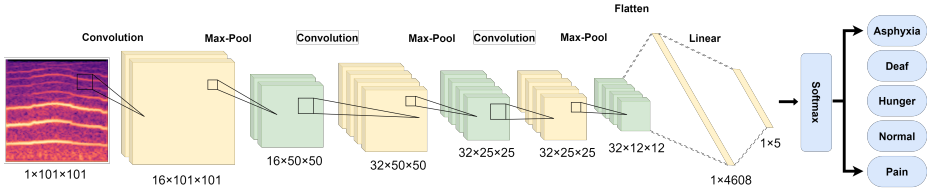

Fig. 2. Generic representation of the CNN architecture

The convolutional component consists of three continuous convolutional layers, each incorporating a 5×5 kernel for feature extraction, followed by a ReLU activation function, and concluded with a maximum pooling operation for downsampling the feature map. These convolutional operations effectively capture significant spatial features from the original audio spectrogram.

The linear component expands the feature graph processed by convolutional and pooling layers into a one-dimensional vector, then goes through a linear transformation, followed by the utilization of softmax activation for classification. The prediction result is determined as the category with the highest probability among the five categories of infant cries.

2.3 CRNN

CRNN is a model that effectively integrates a convolutional neural network (CNN) and a recurrent neural network (RNN) to serve as both feature extractors and classifiers. This architecture has been widely used in DCASE tasks [12, 13]; However, its application in infant cry detection tasks is limited. It is anticipated that leveraging this sophisticated framework can yield promising results for infant cry detection, thereby contributing to advancements in this domain.

In this paper, the CRNN model structure is designed as shown in Fig. 3. The architecture is composed of three main parts: the convolutional component, the recurrent component, and the linear component.

The convolutional component consists of two continuous convolutional layers, each containing a convolutional operation, a Relu activation function, and a maximum pooling operation. The convolution operation uses a 5×5 convolution kernel for feature convolution, while the ReLU activation function introduces nonlinearity to enhance transformation. Additionally, the feature map size is reduced using a 2×2 convolution kernel in the maximum pooling operation.

The recurrent component consists of two stacked GRU units and incorporates dropout regularization. The utilization of GRU cells in this section, instead of

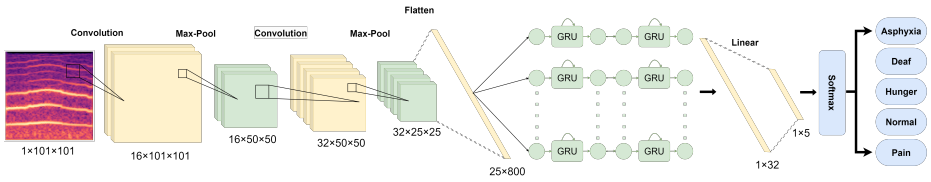

Fig. 3. The CRNN architecture

LSTM cells, is motivated by their reduced parameter requirements while still delivering comparable performance. This operation effectively processes the features obtained by the convolutional layer and captures the temporal dependencies in these features.

The linear component in the proposed CRNN comprises a linear layer that takes the hidden state of the last time step in the GRUs as input and applies a linear transformation to map it to the desired number of output categories. Subsequently, softmax activation is employed to assign probabilities to each class.

2.4 AST

AST, the Audio Spectrogram Transformer [14], which was first proposed by Yuan Gong et.al in 2021, represents a pioneering convolution-free and purely attention-based model for audio classification. Compared with CNN plus attention hybrid model, AST exhibits a simpler structure, fewer parameters, and faster convergence during the training process. The performance of this model is evaluated across various audio classification tasks, all of which demonstrate state-of-the-art results. In [14], the input audio waveform is transformed into a 128-dimensional log mel spectrogram, which serves as the input to the AST. Interestingly, our findings indicate that for infant cry detection, the AST trained using raw spectrograms outperforms the one trained with log mel spectrograms. Consequently, we have decided to utilize raw spectrograms with dimensions of 101×101 as inputs to the AST in order to enhance its performance.

Therefore, in this paper, we change the original AST [14] network structure to our implementation and the details are illustrated in Fig. 4. First, the spectrogram of size 101×101 is divided into N patches of 16×16 overlapping 6 in time dimension and space dimension respectively, where $N = \lceil (101-16)/(16-6) \rceil^2 = 81$. Subsequently, each patch undergoes a linear transformation, resulting in a one-dimensional patch embedding with a dimensionality of 768. To capture the spatial characteristics of the spectrogram, a positional embedding of equal size is added before each patch embedding. Additionally, similar to [14] , we add the [CLS] token at the beginning of the patch embedding sequence. The resulting sequence is then fed into the Transformer encoder with 12 layers, and the output of the [CLS] token serves as feature representations for audio spectrograms. These representations are subsequently labeled and classified using a Multi-Layer Perceptron (MLP) with a sigmoid activation function.

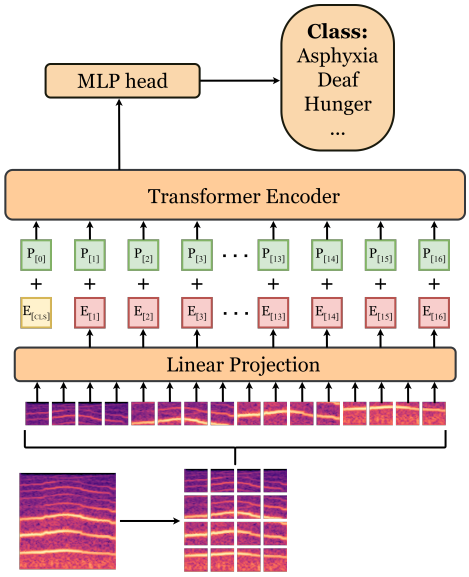

Fig. 4. Details of the AST architecture

By leveraging the concept of transfer learning, we employ the pre-trained weights of ViT [15] on ImageNet [16] to enhance the performance of AST. However, as stated in [14], certain modifications need to be made before implementing transfer learning, due to slight dissimilarities in architecture between ViT and AST. Firstly, since ViT takes a 3-channel image as input while AST requires a single-channel spectrum diagram, it is necessary to average the weights corresponding to the three input channels of ViT's patch embedding layer into a single channel for compatibility with AST. Additionally, we discard the final classification layer of ViT and initialize a new classification layer specifically tailored for AST in order to accommodate disparities between their respective classification tasks.

3 Datasets and Task Design

3.1 Datasets

We use the publicly available Baby Chillanto Database to construct our infant cry detection tasks. This database was collected by the National Institute of Astrophysics and Optical Electronics, CONACYT, Mexico [17]. It consists of five types of infant crying sounds, including asphyxia(340 samples), deaf(879 samples), hunger(350 samples), normal(507 samples) and pain(192 samples). In total, it comprises 2,268 infant cry samples. The cry recordings are captured at sampling rates of 8000Hz, 11025Hz, and 22050Hz, which is shown in Table 1.

We standardizes all wav audio files to a sample rate of 8000 using the librosa library in Python.

The Baby Chillanto Database was recorded under clean condition, however, in this study, we aim to examine the cross-domain generalization ability and noise robustness of three types of DNN-based models. Thus, we introduce noise to the original Baby Chillanto Database by incorporating noise data from Audioset [18]. The Audioset is a comprehensive audio dataset developed by Google for research purposes in audio classification and semantic labeling. This dataset comprises numerous 10-second-long audio clips sourced from YouTube. We locally acquire the Audioset dataset, segmented the audio files into 1-second intervals, randomized their order, and randomly add noise to each individual file within the Baby Chillanto Database. As a consequence, the noisy Baby Chillanto database remains with 2,268 1-second wav files, mirroring the identical count of audio files present in the initial Baby Chillanto database. We utilize this noisy database in noisy related tasks.

Table 1. The summary of Baby Chillanto Database

Label	Number of Samples	Sampling Rate
Asphyxia	340	11025
Deaf	879	8000, 11025, 22050
Hunger	350	8000, 11025, 22050
Normal	507	8000, 11025, 22050
Pain	192	8000, 11025, 22050

3.2 Task Design

In practical application scenarios, the background environment in which a baby's crying occurs is typically very complex, including factors such as people talking, various noises in a household setting like running water, the sound of an electric toothbrush, dish washing noise, mopping sounds, and various noises from the living room TV. Therefore, researching the detection of baby cries in noisy environments is of significant importance for real-world applications.

In order to evaluate the model cross-domain and noise robustness for infant cry detection, we construct three individual training-test tasks in our experiments, they are clean-clean, clean-noise and CNoise-CNoise, respectively. The CNoise means the train and test sets include both the clean and the simulated noisy infant cry segments. The subsequent sections would provide detailed descriptions of the three tasks. The configuration of each task is presented in Table 2.

Clean-Clean task aims to design an infant cry detection system using a clean dataset to divide the samples within the dataset into a ratio of 64:16:20 for training (1451 samples), validation (363 samples), and testing (454 samples). In

Table 2. Task Design

Task Category	Train set	Validation Set	Test set
Clean-Clean	Clean, 1451	Clean, 363	Clean, 454
Clean-Noise	Clean, 1451	Clean, 363	Noisy, 454
CNoise-CNoise	Clean + Noisy, 1451 + 1451	Clean + Noisy, 363 + 363	Clean, 454 , Noisy, 454

this task, our objective is to develop a model capable of accurately detecting and identifying infant cries. We extract audio features from the clean dataset's infant cries and input these features into a deep learning model for modeling and training purposes. The remaining data from the clean dataset serve as both the validation set and test set to validate and evaluate the trained model. This task allows us to assess the accuracy and reliability of the model in an uncontaminated environment. Such results help verify the model's performance under ideal conditions and provide benchmark performance indicators for subsequent tasks.

Clean-Noise task aims to design a cross-domain infant cry detection task where the model is trained using only the clean infant crying segments, while testing using a simulated noisy test set. It is used to validate the robust system capable of effectively identifying infant cries in noisy environments. In contrast to Clean-Clean task, this task necessitates the model's ability to handle diverse real-world noise scenarios. In practical scenarios, the presence of noise is inevitable, necessitating the assurance of proper functionality and sustained accuracy in diverse environments for the infant cry detection system. By assessing the model's performance on a testing dataset encompassing a noisy background, we can gain insights into its real-world efficacy and subsequently enhance or adjust it to optimize application effectiveness while expanding its adaptability.

CNoise-CNoise task aims to design an infant cry detection task that the models are trained using the combination of clean and noisy datasets, while testing on either clean or noisy test set. By amalgamating these two types of data, our objective is to develop a model with exceptional performance and robustness, enabling accurate identification of infant cries across diverse environmental conditions. The integration of clean and noisy conditions in infant cry detection is crucial for establishing a reliable system capable of functioning effectively in various environments. This approach can provide more abundant training data, improve the robustness and adaptability of the model, and make the infant cry detection system more practical and reliable.

To our knowledge, we are the first to explore the performance of infant cry detection under cross-domain and noisy conditions based on the Baby Chillanto Database. We hope these tasks may help to facilitate an in-depth exploration and analysis of infant cry detection under complex scenarios. Moreover, these tasks have laid a solid groundwork for future research endeavors and practical applications in this field.

4 Experiments and Results

4.1 Setups

Features: We convert the audio files from the Baby Chillanto Database into spectrograms with a single channel and a size of 101×101. Fig. 5 showcases a sample spectrogram for each category. We choose to employ spectrograms for infant cry detection is the fact that using spectrograms as input allows the model to capture finer details and preserve the original audio information, thereby minimizing potential information loss.

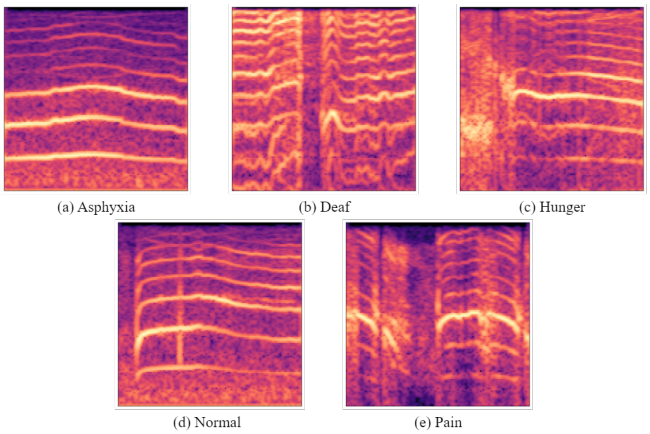

Fig. 5. Spe ctrogram Samples of Baby Chillanto Database

Table 3. Model training configurations

Methods	Epoch	Batch size	Learning rate	Optimizer
CNN	30	64	0.001	Adam
CRNN	50	32		
AST	25	32		

Model configurations: Table 3 presents the detail model training configurations, including the training epoches, batch size, learning rate and optimizer. In Table 4, the specific individual model structure and related parameter settings are listed. For example, the CNN model comprises three convolutional and pooling layers, and one linear layer. The CRNN model comprises two convolutional and pooling layers, two GRU cyclic layers, and one linear layer. The AST

Table 4. Parameters settings of each method

Methods	Parameters
CNN	• Conv1: filter = 16, kernel = 5×5, stride = 1, padding = 2 • MaxPool1: filter = 16, kernel = 2×2, stride = 2, padding = 0 • Conv2: filter = 32, kernel = 5×5, stride = 1, padding = 2 • MaxPool2: filter = 32, kernel = 2×2, stride = 2, padding = 0 • Conv3: filter = 32, kernel = 5×5, stride = 1, padding = 2 • MaxPool3: filter = 32, kernel = 2×2, stride = 2, padding = 0 • Linear: input size = 4608, output size = 5
CRNN	• Conv1: filter = 16, kernel = 5×5, stride = 1, padding = 2 • MaxPool1: filter = 16, kernel = 2×2, stride = 2, padding = 0 • Conv2: filter = 32, kernel = 5×5, stride = 1, padding = 2 • MaxPool2: filter = 32, kernel = 2×2, stride = 2, padding = 0 • GRU1: input size = 800, hidden size = 32, dropout = 0.3 • GRU2: input size = 800, hidden size = 32, dropout = 0.3 • Linear: input size = 32, output size = 5

AST	• Patch Partitioning: patch size = 16, patch number = 81, overlap of time and frequency = 6
	• Patch Embedding: embedding dimension = 768
	• Positional Embedding: embedding dimension = 768
	• Transformer Encoder: encoder number = 12
	  • Layer Normalization
	  • Attention :
	  • Linear(qkv): input size = 192, output size = 576
	  • Dropout
	  • Linear(proj): input size = 192, output size = 192
	  • Dropout
	  • Layer Normalization
	  • MLP :
	  • Linear(fc1): input size = 192, output size = 768
	  • GELU
	  • Linear(fc2): input size = 768, output size = 192
	  • Dropout
	• MLP head:
	  • Layer Normalization
	  • Linear: input size = 192, output size = 5

model partitions the spectrogram input into 81 patches and converts it into one-dimensional patch embedding while incorporating positional embedding of equal size before each patch embedding. Subsequently, the embedding sequence is fed into the Transformer architecture, which encompasses 12 Transformer encoder layers. Each encoder layer consists of a self-attention mechanism and Multi-Layer Perceptron. Finally, a linear layer for classification is attached at the end of the model.

4.2 Results

Table 5 presents the performance evaluations of Clean-Clean, Clean-Noise, and CNoise-CNoise tasks, including the accuracy, precision, recall, and the F1-score [19].

From the results of Table 5, it is clear that the performance under Clean-Clean condition are much higher than those under other cross-domain or noisy-related conditions. Such as, the detection accuracy of three individual models are all above 90%, and the F1-scores are above 85%. Moreover, under the Clean-Clean condition, we find that three heterogeneous models produce similar results in terms of four evaluation metrics. The CRNN and AST only provide slight improvements over the simple CNN model.

Table 5. Accuracy(%), Recall(%), Precision(%) and F1-score(%) on the clean and noisy test sets in three infant cry detection tasks.

Methods	Training Condition	Testing Condition							
		Clean				Noisy			
		Accuracy	Recall	Precision	F1-score	Accuracy	Recall	Precision	F1-score
CNN	Clean	90.09	86.69	84.44	85.26	52.75	51.19	46.14	42.93
CRNN		93.39	89.83	87.75	88.45	52.09	51.94	45.07	41.74
AST		91.41	87.01	84.86	85.50	58.24	**52.72**	41.82	37.34
The Ensemble Model		**94.05**	**92.22**	**89.66**	**90.59**	**59.34**	50.25	**47.94**	**45.87**
CNN	CNoise	88.54	84.11	81.99	82.78	56.04	52.17	48.51	45.70
CRNN		92.29	89.52	86.99	87.94	56.26	59.27	49.22	**46.93**
AST		90.09	85.36	83.93	84.55	58.90	52.29	46.19	43.87
The Ensemble Model		**92.95**	**90.68**	**87.73**	**88.80**	**61.10**	**60.21**	**49.28**	46.65

When comparing the results on the clean and noisy test sets, we see that the performance gap is very large, all the results on the noisy test set are heavily degraded either with the models trained on purely clean or the CNoise samples. This indicates that the performance of current DNN-based infant cry detection techniques deviates far from the industry application requirements, how to improve the noisy robustness of infant cry detection models is fundamental and important for real-world application scenarios.

By comparing the results between Clean-Clean and CNoise-Clean conditions, it's interesting to observe that, the models trained using combination of clean and noisy audio samples don't outperform the models trained on only the clean training samples. The performances are even slightly worse when incorporating the simulated noisy samples into the training data, this may due to these noisy samples introduced an acoustic mismatch between the CNoise and clean test data. However, when comparing the results of CNoise-noisy and Clean-Noisy conditions, the results are significantly improved by adding the noisy training samples to the CNoise training set.

Moreover, from both the results on the clean and noisy test sets, we see that the CRNN outperforms both the simple CNN and complex AST models, it shows better noise robustness and cross-domain generalization ability than other two types of individual models. AST performs comparatively worse than CRNN that

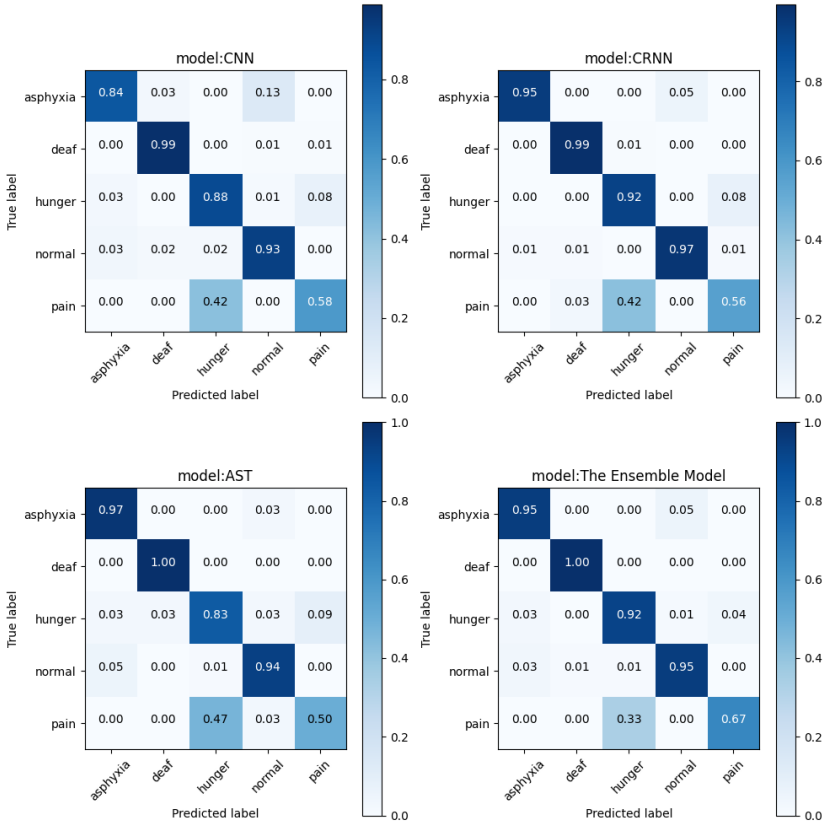

Fig. 6. Normalized Confusion Matrix for different models in the Clean-Clean Task

with simpler structure, possibly because AST encompasses a greater number of parameters, thereby necessitating a larger amount of data for effective training. Furthermore, due to the relatively limited size of the dataset, overfitting may arise during the training phase particularly for complex models like AST.

In addition, when comparing the results on the noisy test set with models trained from clean data and the CNoise data, we find that all the results are significantly improved when we add noise data into the pure clean training set, resulting a better acoustic match between the training and testing conditions.

Finally, it is evident that across three distinct tasks, the ensemble model consistently outperforms all the three individual models. The results clearly highlight the superiority of the ensemble model over individual deep learning models, providing compelling evidence for the complementary information between different models with different networks structures.

Figure 6 illustrates the confusion matrices for each model in the Clean-Clean task. As evident from the confusion matrices, the models face challenges when

attempting to differentiate between hunger infant cries and pain infant cries. This difficulty can be attributed to the significant similarity in the spectrograms of hunger and pain infant cries, making it challenging for the model to distinguish their differences. It's worth noting that the confusion between the other four types of infant cries is minimal, demonstrating a high level of discrimination among these cry types. Furthermore, by comparing the confusion statistics between the ensemble model and three individual models, we find that the complementary information fusion can greatly alleviate the acoustic confusion between hunger and pain infant cries, thus bring additional performance gains over all of the individual models.

5 Conclusion

In this paper, we explored three distinct DNN-based models, namely CNN, CRNN, and AST, for infant cry detection. Moreover, we introduced the ensemble learning as a means to combine these three individual models, leveraging their complementary information. To evaluate the model noise robustness and cross-domain generalization ablity, we designed three distinct tasks, each with varying training and testing conditions to see the behavior of each types of DNN-based models. The experimental findings demonstrated that the three heterogeneous DNN-based models exhibited varying advantages under different infant cry detection conditions across the three distinct tasks. Importantly, ble model effectively exploited the complementary information from these individual models, thus resulting better detection performances in terms of variety evaluation metrics. Our future work will focus on the techniques for improving the infant cry detection performance under complex noisy scenarios.

References

1. O. Wasz-Höckert, T.J. Partanen, V. Vuorenkoski, and et. al. The identification of some specific meanings in infant vocalization, Experientia, 20: 154, 1964.
2. C.Y. Chang, Y.C. Hsiao, and S.T. Chen. Application of incremental SVM learning for infant cries recognition, in International Conference on Network-Based Information Systems, 2015, pp. 607-610.
3. S. Bano, and K.M. RaviKumar. Decoding baby talk: A novel approach for normal infant cry signal classification, in Proc. ICSNS, 2015, pp. 1-4.
4. I.A. Bănică, H. Cucu, A. Buzo, and et. al. Automatic methods for infant cry classification, in Proc. COMM, 2016, pp. 51-54.
5. S.F. Molaeezadeh, and M. Salarian. Type-2 fuzzy pattern matching for classifying hunger and pain cries of healthy full-term infants, in Proc. AISP, 2012, pp. 233-237.
6. H.N. Ting, Y.M. Choo, and A.A. Kamar. Classification of asphyxia infant cry using hybrid speech features and deep learning models, Expert Systems with Applications, 208: 118064, 2022.
7. A. Gorin, C. Subakan, S. Abdoli, and et. al. Self-supervised learning for infant cry analysis, arXiv preprint arXiv:2305.01578, 2023.

8. J. Xu, W. Wang, H. Wang, and et. al. Multi-model ensemble with rich spatial information for object detection, Pattern Recognition, 99: 107098, 2020.
9. X. Chen, Y. Sun, B. Athiwaratkun, and et. al. Adversarial deep averaging networks for cross-lingual sentiment classification, Transactions of the Association for Computational Linguistics, 6: 557-570, 2018.
10. E. Franti, I. Ispas, and M. Dascalu. Testing the universal baby language hypothesis-automatic infant speech recognition with cnns, in Proc. TSP, 2018, pp. 1-4.
11. Y.C. Liang, I. Wijaya, M.T. Yang, and et. al. Deep learning for infant cry recognition, International Journal of Environmental Research and Public Health, 19(10): 6311, 2022.
12. H. Jallet, E. Cakır, and T. Virtanen. Acoustic scene classification using convolutional recurrent neural networks, the Detection and Classification of Acoustic Scenes and Events (DCASE), pp. 1-5, 2017.
13. D. Wang, L. Zhang, C. Bao, and et. al. Weakly supervised CRNN system for sound event detection with large-scale unlabeled in-domain data, arXiv preprint arXiv:1811.00301, 2018.
14. Y. Gong, Y.A. Chung, and J. Glass. Ast: Audio spectrogram transformer, arXiv preprint arXiv:2104.01778, 2021.
15. H. Touvron, M. Cord, M. Douze, and et. al. Training data-efficient image transformers & distillation through attention, in Proc. PMLR, 2021, pp. 10347-10357.
16. J. Deng, W. Dong, R. Socher, and et. al. Imagenet: A large-scale hierarchical image database, in IEEE conference on computer vision and pattern recognition, 2009, pp. 248-255.
17. O.F. Reyes-Galaviz, S.D. Cano-Ortiz, and C.A. Reyes-García. Evolutionary-neural system to classify infant cry units for pathologies identification in recently born babies, in IEEE Seventh Mexican international conference on artificial intelligence, 2008, pp. 330-335.
18. J.F. Gemmeke, D.P.W. Ellis, D. Freedman, and et. al. Audio set: An ontology and human-labeled dataset for audio events, in Proc. ICASSP, 2017, pp. 776-780.
19. C. Goutte, and E. Gaussier. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation, in European conference on information retrieval, 2005, pp. 345-359.