Hybrid ViT-Based Sleep Sound Classification

Description

This repository contains all code and materials for a deep learning-based sleep sound classification study using hybrid ViT + CNN + BiLSTM + Attention architectures. We process audio signals into spectrogram images (Mel, MFCC, CQT), apply SpecAugment for robustness, and utilize ensemble learning (average, weighted, stacking) to improve accuracy. The final model performs 10-fold cross-validation and is benchmarked against multiple ensemble strategies.

Dataset Information

- **Input**: 700 sleep sound samples in .wav format, 48kHz, 4–8 seconds each, across 7 classes.
- **Class Labels**:
- 0: Cough, 1: Laugh, 2: Scream, 3: Sneeze, 4: Snore, 5: Sniffle, 6: Farting
- **Spectrograms Used**:
  - Mel Spectrogram
  - MFCC
  - CQT

Original audio files are stored under: SLEEP/nocturnal_wav/

Spectrogram images are saved to: SLEEP/mel/, SLEEP/mfcc/, SLEEP/cqt/

Code Information

File	Description
1-ensemble_from_wav_multibranch_regularization.py	Extracts spectrograms, trains hybrid models on each branch, performs 10-fold CV
2-save_branch_predictions.py	Loads trained models and saves softmax outputs (y_prob) for each fold
3-evaluate_branches_10fold.py	Evaluates each branch model and plots confusion matrices
4-weighted_ensemble_evaluation.py	Performs weighted softmax ensemble based on individual model accuracies

File	Description
5-stacking_logreg.py	Applies Logistic Regression stacking using concatenated softmax vectors
6-stacking_xgboost.py	Applies XGBoost stacking for final predictions
7-confidence_distribution.py	Visualizes softmax confidence histograms for each model branch
8-pca&tsne.py	Visualizes softmax vectors using PCA and t-SNE

Requirements

Install required Python packages with:

```
pip install -r requirements.txt
```

Methodology

- Each .wav file is converted into 3 spectrogram types.
- A **hybrid model** combining CNN + BiLSTM + Attention + ViT is trained per branch.
- **SpecAugment** is applied during training for augmentation.
- **10-fold stratified cross-validation** is used for evaluation.
- Final classification is conducted using:
  - Average Softmax Ensemble
  - Weighted Ensemble
  - Stacking (LogReg and XGBoost)

Evaluation Method

- **Cross-validation**: 10-fold stratified
- **Comparative Baselines**: Individual branches, average ensemble, weighted ensemble, stacking
- **Visualization**: Confusion matrices, ROC curves, t-SNE, PCA

Assessment Metrics

The following metrics are used and reported for each experiment:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**
- **AUC (macro average)**

Usage Instructions

To run a full training and evaluation pipeline:

```
python 1-ensemble_from_wav_multibranch_regularization.py
python 2-save_branch_predictions.py
python 3-evaluate_branches_10fold.py
python 4-weighted_ensemble_evaluation.py
python 5-stacking_logreg.py
python 6-stacking_xgboost.py
python 7-confidence_distribution.py
python 8-pca&tsne.py
```

Outputs (e.g., .npy, .png, .txt) are saved in:

```
SLEEP/results_ensemble_multibranch/
```

Citations

If you use this repository, please cite the associated research article:

Sağbaş E.A., Nocturnal sleep sound classification with multi-spectrogram feature fusion and an attention-based stacked hybrid ConvBiLSTM-ViT architecture, PeerJ Computer Science, Under Review.