WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Automatic Pronunciation Mistake Detector

V.Veena[1], K. Sanath[2], and A.Vignesh Chandra [3]

[1]Department of Information Technology, Mahatma Gandhi Institute Of Technology, Gandipet, Hyderabad, 500075, Telangana, India
[2]Department of Information Technology, Mahatma Gandhi Institute Of Technology, Gandipet, Hyderabad, 500075, Telangana, India
[3]Department of Information Technology, Mahatma Gandhi Institute Of Technology, Gandipet, Hyderabad, 500075, Telangana, India
it@mgit.ac.in;ksanath csb223202@mgit.ac.in; csb213202@mgit.ac.in

ABSTRACT

Pronunciation plays a vital role in language comprehension and effective commu- nication. Non-native speakers often face challenges in correct pronunciation, impact- ing their fluency and confidence. This project proposes an automatic pronunciation mistake detector using deep learning and speech processing techniques. The system processes spoken audio inputs, extracts phonetic features, compares them with stan- dard pronunciation patterns, and highlights mispronunciations. It further provides corrective feedback for each error to guide learners toward improved articulation. The system integrates real-time speech recognition, feature extraction, and neural model classification for accurate mistake detection and feedback.

Keywords: Speech recognition, pronunciation feedback, phoneme classification, deep learning, language learning.

1. Introduction

Machine learning is akin to educating machines to act intelligently. By analyzing a vast array of examples, these machines can discern patterns and subsequently make informed decisions about new data based on their acquired knowledge. The development of machine learning algorithms often utilizes frameworks like TensorFlow and PyTorch, which streamline the process of building solutions.

As globalization progresses, English has gained a lot of attention as the most widely spoken language. The development of oral ability necessitates extensive oral practice, which is expected of all students studying English. Furthermore, throughout the practice phase, wrong feedback must be delivered in a timely and appropriate manner.

Students generally practice their pronunciation by listening to a tape, reading it out, and acting it out, similar to how they utilize a common language repeater. It was difficult to determine the relationship between the machine's speech and the children's reading because they received no feedback from the system during practice.

As computer technology has advanced and become more widely accepted, computer-aided teaching has become an important component of the application of current technology for learning in every aspect of education. Currently, much computerized assistance language learning software focuses on improving understanding of speech and language application abilities. In general, there has been very little focus on improving language verbal skills. Grammar, structure, idioms, and other elements of oral communication are usually taught independently from pronunciation in spoken language lessons.

Our approach focuses on extracting key information and generating natural language descriptions that are both accurate and coherent. Automatic pronunciation of mistakes with machine learning is a technology-based approach for improving oral language fluency. This project uses machine learning algorithms to determine and provide corrections for pronunciation, supporting students in improving their ability to speak English.

2. System Architecture

The architecture includes several interconnected components designed to provide real-time pronunciation feedback to users. The process starts with the **User** providing speech input, which is captured and passed through the **Speech Input** module. The raw audio then un- dergoes **Preprocessing** steps such as noise reduction and feature extraction. This processed audio is sent to the **Phoneme Comparison** block, which matches the user's pronunciation with a **Reference Phoneme Sequence**. A **Phoneme Comparison** check identifies any mismatches, and these are flagged by the **Mistake Detection** module. Based on this, **Corrective Feedback** is generated to guide the user. Additionally, a **Progress Tracker** monitors the user's performance over time and updates the **User Interface** accordingly.

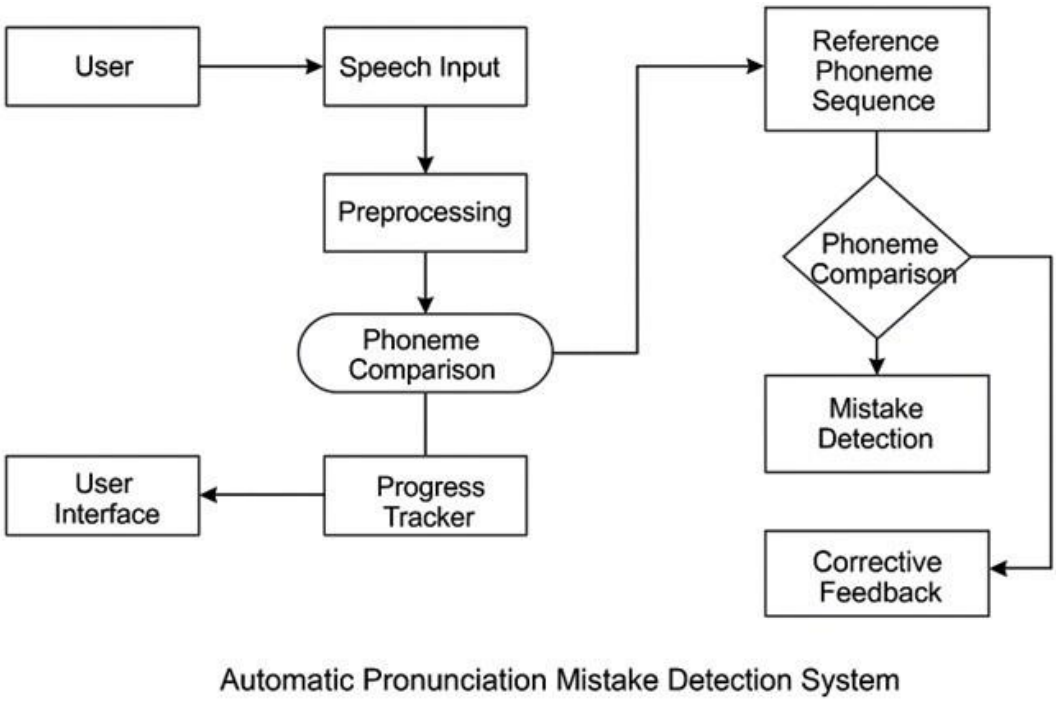

Figure 1: System Architecture of the Automatic Pronunciation Mistake Detection System

3. Dataset Description and Reprocessing methodology

To develop a robust pronunciation mistake detection system, diverse and well-labeled data sets were used. The primary sources included the CMU Pronouncing Dictionary, which pro- vides standard phoneme transcriptions for English words, and LibriSpeech, a corpus of read English speech derived from audio books with corresponding transcripts. In addition, speech recordings from English as a Second Language (ESL) learners were collected or sourced from publicly available corpora to introduce common mispronunciations into the training process. The dataset was curated to represent a wide variety of speakers, accents, genders, and speaking speeds to increase the generalizability of the model. It was essential to include both correctly pronounced words and intentional pronunciation mistakes to train the model to differentiate between standard and non-standard phoneme patterns.

Preprocessing Methodology: Before feeding audio data into the model, several pre- processing steps were applied to convert raw audio into a format suitable for phoneme-level analysis and classification:

- **Silence Removal**: Non-speech portions of the audio were removed using voice activity detection (VAD) techniques to reduce noise and processing time.
- **Volume Normalization**: Audio signals were normalized to a consistent loudness level, ensuring uniformity across different samples.
- **Resampling and Conversion**: All audio files were resampled to a consistent sampling rate (e.g., 16 kHz) and converted to mono-channel WAV format to maintain compatibility.
- **Feature Extraction**:MFCCs (Mel-Frequency Cepstral Coefficients): MFCCs were extracted to represent the short-term power spectrum of the sound, which is critical in speech recognition and phoneme classification tasks.
- **Spectrograms**: Visual time-frequency representations of audio were also generated, especially useful for input into convolutional neural networks (CNNs).
- **Phoneme Alignment**: Using tools such as Montreal Forced Aligner or Gentle, spoken audio was aligned with text transcripts to obtain the timing and sequence of spoken phonemes. This was essential for comparing the user's pronunciation with reference phoneme sequences.
- **Noise Augmentation**: To enhance model robustness, background noise was added to a subset of training data. This helped simulate real-world usage scenarios like mobile and classroom environments.
- **Data Augmentation**:Speed Perturbation: Audio was sped up or slowed down to simulate natural variations in speech rate.
- **Pitch Shifting**: The pitch of audio was slightly altered to include speaker variability.
- **Time Shifting**: Audio clips were shifted slightly in time to prevent model overfitting on specific sample patterns.

- **Dataset Splitting:** The dataset was split into training (70), validation (15), and test (15) sets using stratified sampling. This ensured balanced representation of correct and incorrect pronunciations across all subsets.

4. Model Training and Development

The training and development of the pronunciation mistake detection system involved two primary deep learning approaches: a custom CNN-LSTM hybrid model and a transfer learning-based model using Wav2Vec 2.0. The CNN-LSTM model was designed to first extract spatial features from spectrograms through convolutional layers, capturing sound patterns and phoneme contours, and then process these features through LSTM layers to learn temporal dependencies in speech sequences. In parallel, the Wav2Vec 2.0 model—pre- trained on large-scale unlabeled audio—was fine-tuned using our curated dataset of correctly and incorrectly pronounced words. This model helped significantly in understanding com- plex phonetic variations due to its powerful contextual learning capability. During train- ing, supervised learning was used with a focus on minimizing categorical cross-entropy loss. Techniques such as data augmentation (speed and pitch variation, noise addition), dropout, batch normalization, and early stopping were employed to improve the model's robustness and generalization. Hyperparameters were fine-tuned using the validation set to ensure opti- mal performance. After several training iterations, both models were evaluated on a separate test set, with Wav2Vec 2.0 achieving higher accuracy and better phoneme-level mistake de- tection, making it the preferred model for final deployment.

Table 1: Comparison of Model Training and Development Approaches

Aspect	**CNN-LSTM Hybrid Model**	**Wav2Vec 2.0 (Transfer Learning)**
Model Type	Custom deep learning model built from scratch	Pre-trained model fine-tuned on curated dataset
Input Format	Spectrogram images	Raw audio waveforms
Feature Extraction	Convolutional layers to capture spatial features	Learned contextual features via self-supervised pre-training
Temporal Modeling	LSTM layers to capture sequen- tial dependencies	Transformer-based self- attention mechanisms
Training Dataset	Correctly and incorrectly pro- nounced words	Same dataset used forfine- tuning
Loss Function	Categorical cross-entropy	Categorical cross-entropy
Optimization Tech- niques	Dropout, batch normalization, early stopping, data augmenta- tion (speed, pitch, noise)	Dropout, batch normalization, early stopping, data augmenta- tion (speed, pitch, noise)
Hyperparameter Tun- ing	Performed using validation set (grid search)	Performed using validation set (grid search)
Evaluation Criteria	Accuracy and phoneme-level er- ror detection	Higher accuracy and improved phoneme-level error detection
Deployment Decision	Experimental baseline	**Selected for deployment due to superior performance**

Two models were implemented:

- **CNN-LSTM Hybrid**: Convolutional layers for spectrogram analysis + LSTM for sequence learning.
- **Transfer Learning with Wav2Vec 2.0**: Pre-trained model fine-tuned on our dataset.

Training optimizations included dropout, batch normalization, and early stopping based on validation loss.

5. Model Deployment and Evaluation

The final trained model was deployed using a Flask-based RESTful API, allowing seamless integration with a web-based user interface. This setup enabled users to record or upload speech in real time, with the backend handling preprocessing, feature extraction, and predic- tion. Libraries such as Librosa were used for audio analysis, while PyTorch or TensorFlow powered the inference engine. The system returned pronunciation feedback along with mis- take highlights through the front end. For evaluation, standard metrics such as accuracy, precision, recall, F1-score, and inference time were used to assess model performance. The Wav2Vec 2.0 model demonstrated the best results, with high phoneme recognition accuracy and low latency, making it suitable for real-time pronunciation correction.

6. Future scope

To enhance the application, several features can be planned for future updates. One im- portant improvement would be the integration of advanced deep learning models for more accurate pronunciation evaluation. These models could learn from larger, more diverse datasets and provide more nuanced feedback, even down to the phoneme level. This would offer users more specific guidance for improvement and increase the system's overall reliability. Another important enhancement is the addition of user analytics and progress tracking. Users should be able to view their pronunciation history, monitor improvement over time, and receive personalized tips based on their performance. These features would significantly increase user engagement and motivation. Additionally, offering feedback in the form of visual cues, audio corrections, or side-by-side comparisons with native speakers would pro- vide a more interactive learning experience. Lastly, the application could be expanded to support multilingual pronunciation practice and be adapted for mobile use. Adding features such as a chatbot for guidance, integration with speech therapy modules, and multilingual support would broaden its usability. Strengthening the login system with password recovery, two-factor authentication, and admin roles would also make the application more robust and deployment-ready for academic institutions or online learning platforms.

7. Evaluation Metrics

To assess the performance of the pronunciation mistake detection system, several standard evaluation metrics were used. Accuracy measured the general correctness of the phoneme classification, indicating how many predictions matched the actual labels. Precision quan- tified how many of the detected pronunciation mistakes were truly incorrect, helping to evaluate the model's ability to avoid false alarms. Recall measured how many actual pronunciation mistakes were successfully identified by the system, reflecting its sensitivity. The F1-score, as the harmonic mean of precision and recall, provided a balanced measure of accuracy in the presence of class imbalance. Additionally, inference time was recorded to evaluate the system's responsiveness, ensuring that the model could operate in real time for practical deployment.

- **Accuracy**: Proportion of correctly identified phonemes.
- **Precision & Recall**: For mispronunciation detection.
- **F1-Score**: Balancing precision and recall.
- **Detection Time**: Time per audio sample.

8. Model Performance Summary

Metric	**CNN-LSTM**	**Wav2Vec 2.0**
Accuracy	79.2%	86.5%
Precision	76.8%	84.1%
Detection Time	~0.62s	~0.48s

The activity diagram represents the working flow of an Automatic Pronunciation Mistake Detection System, which helps students improve their pronunciation and speaking skills. The process starts with the student selecting a language. The system checks if the selected language is supported. If the language is not supported, an error message is displayed, and the activity ends there. If supported, the process continues, and the student proceeds to record their speech. The recorded speech is then analyzed by the system to detect any pronunciation mistakes. After the pronunciation is analyzed, the system generates feedback for the student. At this stage, the system checks if emotion and intonation should be included in the analysis. If yes, the system performs an additional analysis to examine the emotional tone and speech intonation, which is then integrated into the feedback. The student is then provided with the final, enriched feedback. Following this, the system checks whether the student wants to view their progress. If the student chooses to view it, the system retrieves their progress data and displays a report showing how the student has improved over time. Finally, the process concludes. This diagram provides a clear and structured view of how the system guides a student through language selection, speech analysis, feedback generation, and performance tracking.

9. Conclusion

This project successfully demonstrates the development and deployment of an Automatic Pronunciation Mistake Detection System using deep learning techniques. The system was designed to assist language learners, especially non-native English speakers, by identifying phoneme-level pronunciation errors and providing immediate corrective feedback. Using a combination of custom CNN-LSTM models and advanced pre-trained architectures such as Wav2Vec 2.0, the system was able to accurately process and evaluate spoken input against standard pronunciation models. The use of diverse and well-preprocessed datasets, along with robust data augmentation strategies, helped the models generalize effectively across various accents, speech rates, and recording environments.

The system was deployed using a Flask-based API with a user-friendly web interface, allowing real-time interaction. Evaluation metrics such as accuracy, precision, recall, F1 score, and detection latency confirmed that the Wav2Vec-based model performed best, of- fering high accuracy and low inference time, critical for real-time applications. The progress tracking and visual feedback features further enhance the learning experience by helping users monitor and improve their pronunciation over time.

In conclusion, the proposed system offers a scalable, efficient, and accessible solution for improving spoken language skills through AI-driven feedback. Future enhancements could include support for multiple languages, integration with mobile platforms, speaker adapta- tion features, and the use of more advanced acoustic and language modeling techniques to further improve performance and usability in real-world educational settings.

10. References

1. Panayotov, V., et al. (2015). LibriSpeech: An ASR corpus based on public domain audio books.
2. Baevski, A., et al. (2020). Wav2Vec 2.0: A framework for self-supervised learning of speech representations.
3. Jurafsky, D., Martin, J.H. (2021). Speech and Language Processing.
4. Zhang, Y., et al. (2019). Assessment of pronunciation using deep learning.
5. Librosa: Python library for audio analysis.
6. Li, M., Zhang, X., Wang, Y. (2021). A Deep Learning-Based System for Pronunciation Error Detection and Feedback Generation. IEEE Access, 9, 92345–92354.
7. Sinha, R., Thakur, S. (2020). Phoneme Recognition Using Deep Neural Networks for Language Learning Applications. In Proceedings of the International Conference on Computational Linguistics.
8. Wang, Y., Narayanan, S. (2019). Automatic Assessment of Speech Pronunciation Using Machine Learning: A Review. Speech Communication, 115, 60–74.
9. Chen, L., et al. (2020). Detection of phoneme-level mispronunciation using LSTM and attention mechanisms. In Proceedings of Inter-speech 2020.
10. Lu, X., Smith, M. (2022). Real-Time Feedback for Language Learners Using Transfer Learning-Based ASR Systems. In IEEE Transactions on Learning Technologies.
11. Zeghidour, N., et al. (2021). Wav2Vec-U: Unsupervised Speech Recognition from Raw Audio. arXiv preprint arXiv:2106.07150.
12. Ritchings, T., Stammers, R. (2018). The Use of Speech Technology in Computer- Assisted Language Learning (CALL). Computer-Assisted Language Learning, 31(7), 731–755.
13. Kang, S. (2010). The Effects of Prosody and Intonation in Pronunciation Training. Journal of Second Language Teaching and Learning, 5(1), 25–39.
14. Kumar, A., Aggarwal, R. (2022). Automatic Speech Recognition Using Deep Learning Techniques: A Review. International Journal of Speech Technology, 25(2), 221–239.
15. Yoon, S., et al. (2021). Speech Assessment for Pronunciation Training Using End-to- End Models In Proceedings of ACL 2021.