Real-Time Transcription and Speaker Recognition from Multi-Speaker Audio

Connor Boyd-Lyon

MSc Data Science

Supervised by Dr. Phil Smith

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2024-25

Abstract

Acknowledgements

Abbreviations

ACB	Apple Banana Carrot

Contents

List of Figures

List of Tables

CHAPTER 1

Methodology

The Methodology covers the process for training the Speaker Count CNN and pushing the model to production. First the data collection process is described, followed by the pre-processing required to create the supervised data set of fixed duration voice clips, their spectrograms and the supervision labels. Next, the model architecture is described. Once the architecture was defined and the dataset was prepared, hyperparameter tuning was done with Optuna to find the optimal model hyperparameters, including the number of convolutional layers, skip-connections, optimiser and kernel size. Finally, the method is given for sending the model to a production environment, and how the model was prepared for usage in real-time.

1.1 Data Collection

The dataset used in model training, testing and validation is LibriCSS (cite), a corpus specifically designed to evaluate continuous speech separation systems. LibriCSS is a re-recorded and modified version of the LibriSpeech (cite) dataset. LibriCSS was created by playing LibriSpeech audio through speakers in a real room and recording playback through multiple channels. This means that spatial information and noise was captured; LibriCSS reflects real-world distortion in its data, which is crucial for a robust model post-production. LibriCSS contains 682 distinct speakers, whose utterances were concatenated at various levels of overlap (Table 1.1).

Table 1.1: The types of audio present in LibriCSS (cite). Training a model on all 6 levels of overlap gave the best generalisation.

Audio Type	**Meaning**
0L (zero-long)	No overlap, inter-utterance gap of 3 seconds.
0S (zero-short)	No overlap, inter-utterance gap of 0.5 seconds.
OV10 (overlap-10)	10% overlap on utterances.
OV20 (overlap-20)	20% overlap on utterances.
OV30 (overlap-30)	30% overlap on utterances.
OV40 (overlap-40)	40% overlap on utterances.

1.2 Data Pre-Processing

Why LibriCSS Is Realistic LibriCSS isn‘t just synthetic—it‘s physically re-recorded. The process involves: - Playing LibriSpeech audio through speakers in a real room - Recording the playback via multiple microphones, simulating a multi-channel, spatial environment This setup captures acoustic phenomena like: - Room reverberation - Spatial separation between speakers - Microphone distance/angle variation So instead of sterile waveform superposition, LibriCSS reflects real-world distortions and interferences—crucial for robust model evaluation.

How Mono Input Ties Into This Your CNN only receives mono inputs, so: - You‘re essentially flattening spatially rich data into a single-channel stream - This mimics deployment settings: one mic, messy environment, real-time prediction Framing it this way shows that you’re not discarding spatial information arbitrarily—you‘re doing it intentionally, to develop a model that thrives even without multi-mic setups.

Although LibriCSS is recorded using multiple microphones to simulate spatially rich audio scenes, our speaker count model accepts only mono input. This flattening of spatial information reflects real-world constraints, where predictions must be made from single-channel audio. By training on spatially recorded but mono-processed data, the model learns to infer speaker counts robustly—even without access to spatial cues.

1.3 Model Architecture

1.4 Hyperparameter Tuning with Optuna

1.5 Preparing the Model for Real-Time Evaluation

1.6 Pushing the Model to Production

Bibliography