**Preprocessing**

The videos are fed into a preprocessing pipeline that is outline in the following major steps:

1. Audio is extracted from the video.
2. STFT noise reduction is done.
3. A sequence of MFCC features is extracted from the audio signal at given framing windows(specified by frame\_length and hop\_length)
4. The extracted MFCC sequence is down sampled into a fixed sequence length to ensure constant input shape for the neural networks.

**Model Training**

After preprocessing, the sampled MFCCs are fed into RNNs to process the audio signals and find important temporal cues that might be good indicators of lying. The best performing model was a bidirectional LSTM based model.

**Important remarks and discoveries**

Some approaches promote splitting every single audio signal into several ones and compute MFCC of each, but these approaches are not accurate because it is not clear which part of the clip displayed deception cues. (although this approach does act as some sort of augmentation method, but its results are illogical and inaccurate)

Other approaches were tried like passing the mean MFCC to ANNs or ML Classifiers like SVMs and Random Forests, and those seemed to yield good results with accuracies reaching 80% and 85% as stated in some papers (and as we achieved with our trials) but on manually splitting the data to ensure no bias towards subjects, these same models accuracy was much worse achieving results around 50%, this suggests that the previous attempts that achieved accuracies of up to 85% weren’t really detecting lies.

Our approach was to pass a sequence of MFCCs (rather than mean MFCCs) to an RNN and this approach yielded 79% accuracy on a non-biased manual split.