SPARTANS WILL.

Week15
Preliminary Results

Hanqing Guo

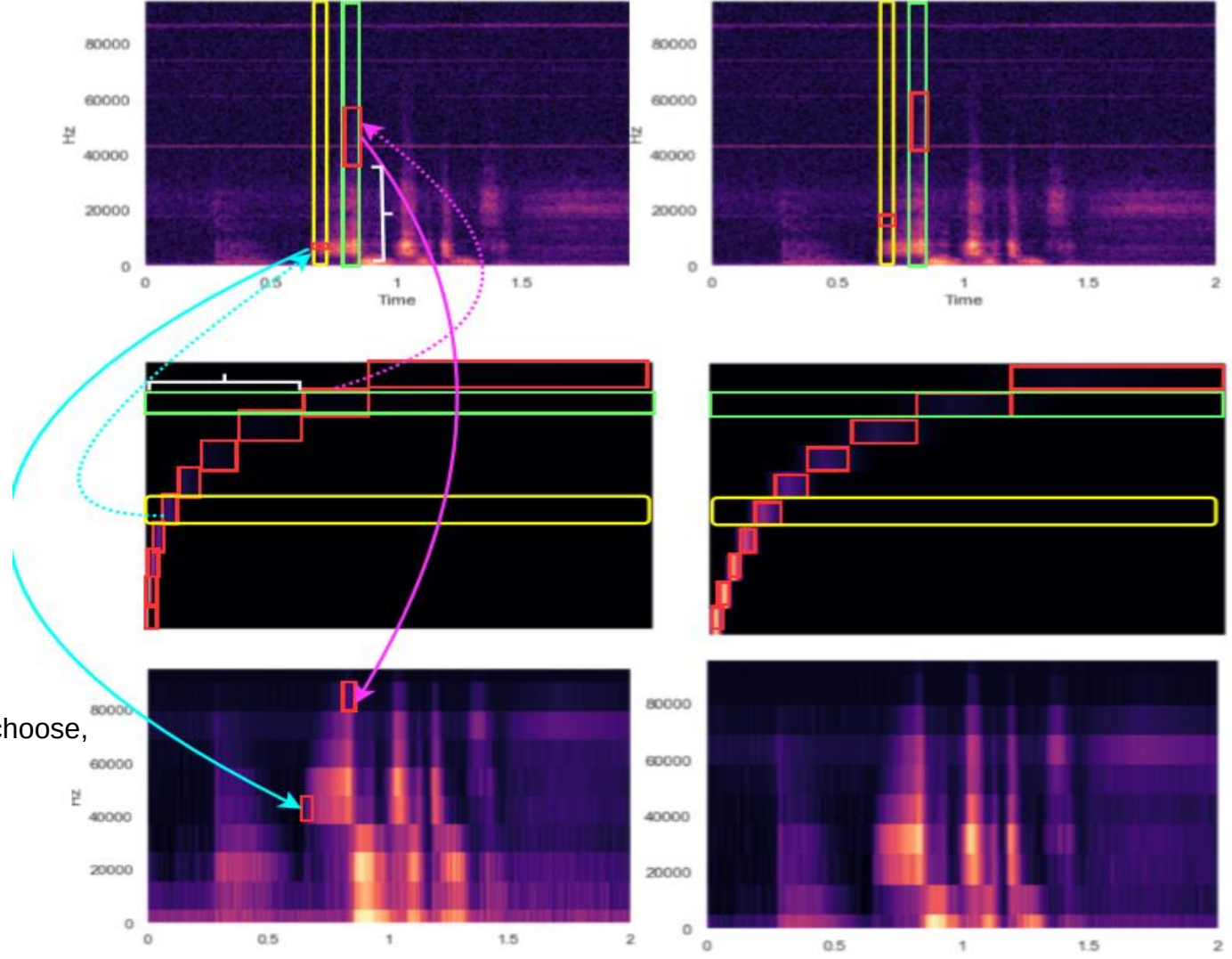
Content

- 1. Quick Review the Benefits of Ultrasound
- 2. Review 4 Different Methods: sliding window, Pattern match, GMM-UBM, Google
- 3. Result Comparison
- 4. Conclusion

SPARTANS WILL.

Quick Review the Benefits of Ultrasound

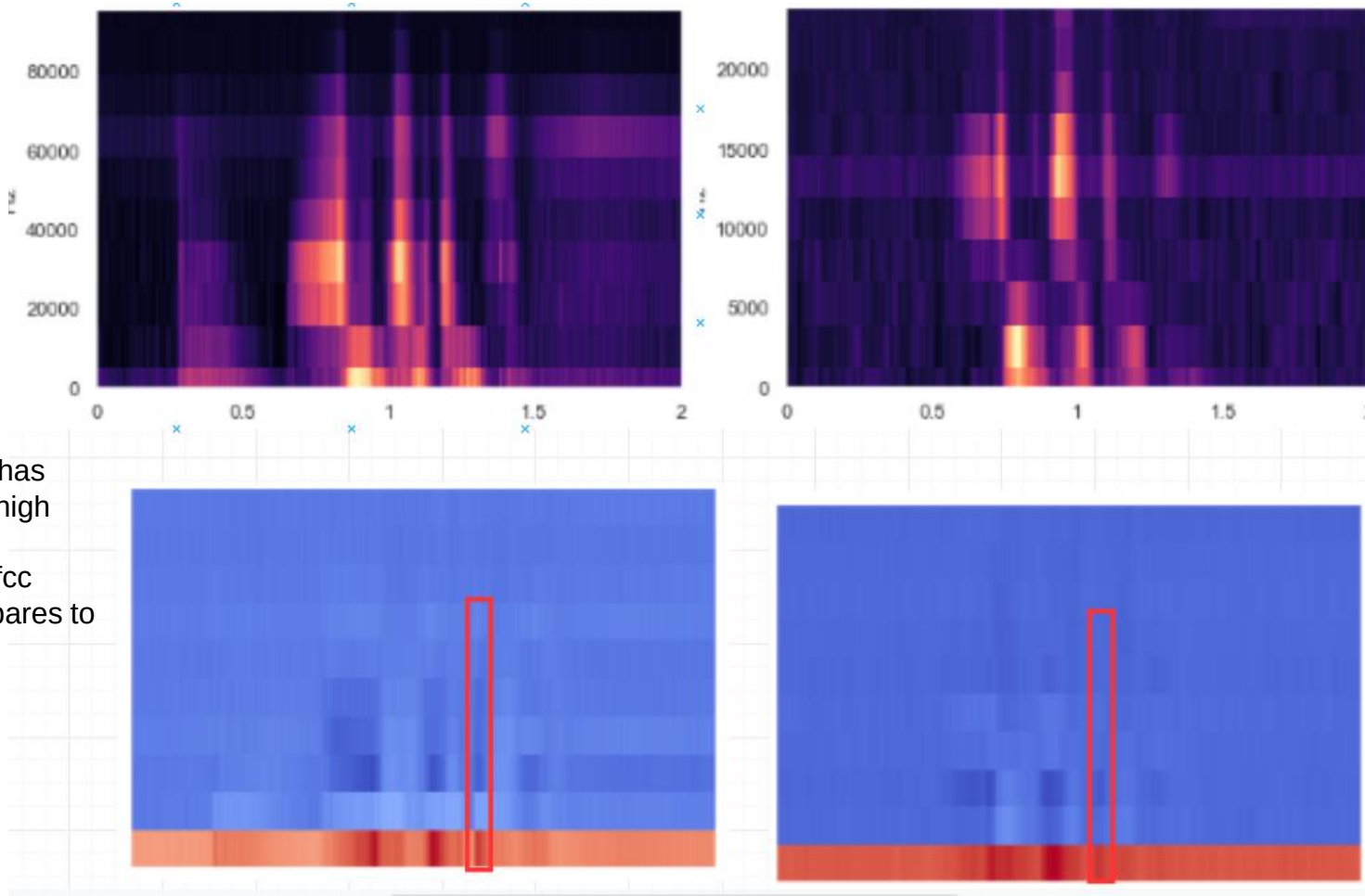
Different Scale Mel-filters



Conclusion1:
Proper mel-filter banks will expose more high-frequency Feature

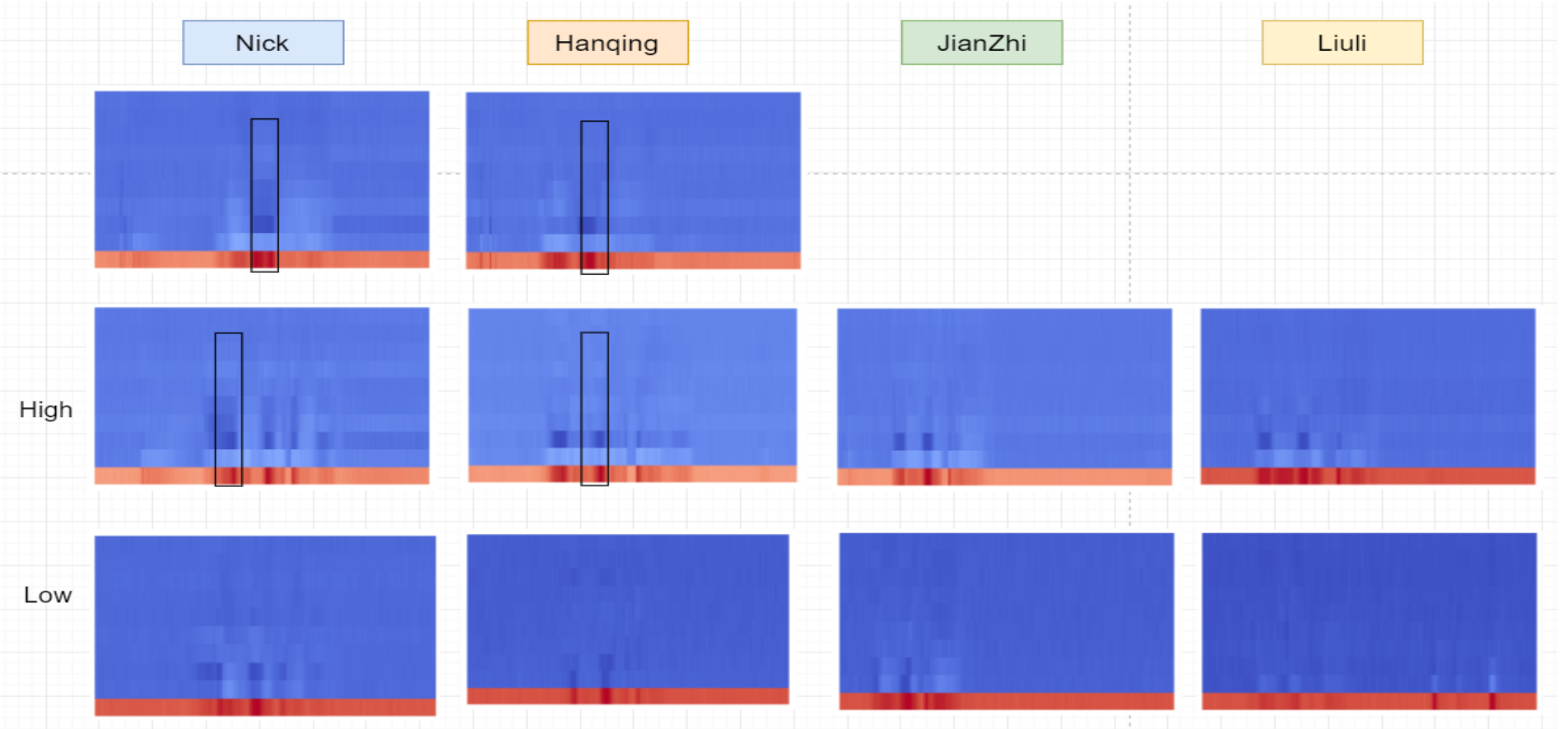
Conclusion2:
Whatever mel-filter number choose, It will represent all frequency range feature

High vs Low raw data and MFCC



Conclusion3:
Ultrasound microphone data has
- enriched feature because high time resolution
- the continuous identical mfcc features are more clear compares to normal microphone.

Is this MFCC identical? It seems Yes, but we will see later



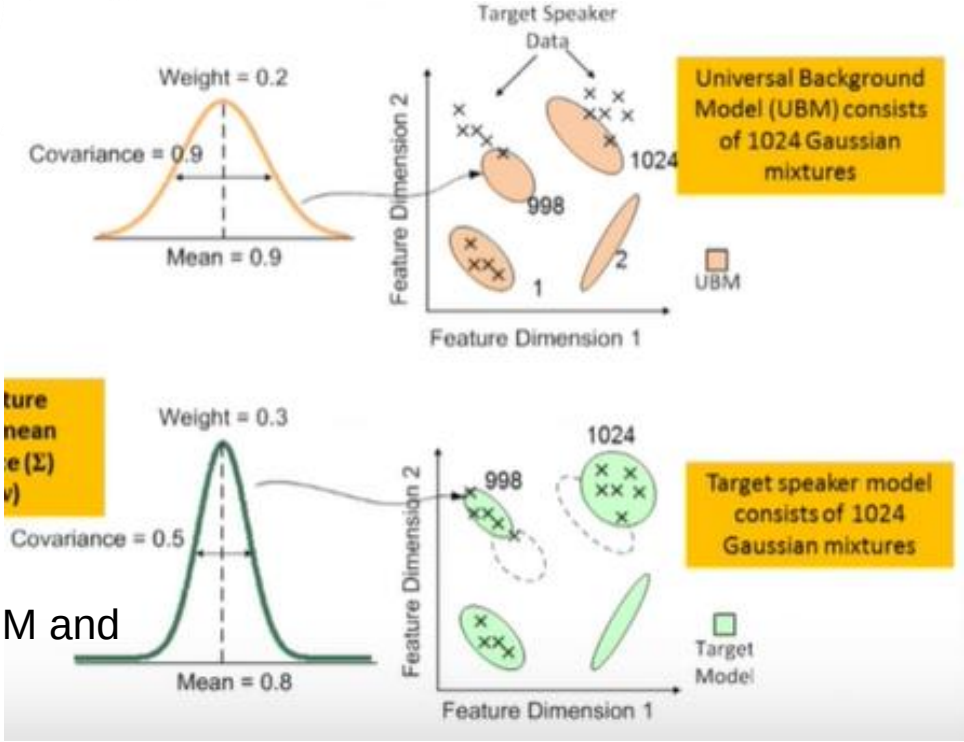
SPARTANS WILL.

Review four Methods

Gaussian Mixture Model

To verify 10 speakers:
1,2,3,4,5,6,7,8,9,10

Train Universal Background Model first. Because every speaker needs normally 1024 mixtures, while the 10 speakers only enroll limited utterance, they are not able to fit many mixtures.



To train UBM, need other 50 speakers, each Speakers collects longer utterance.

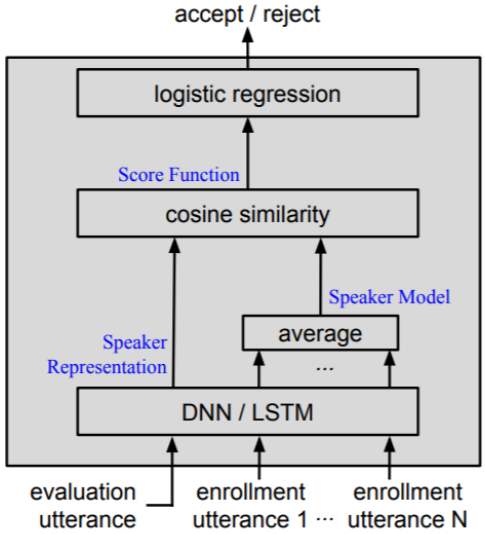
Then for each speaker to be verified, fit the UBM and Build 1024 mixtures for everyone.

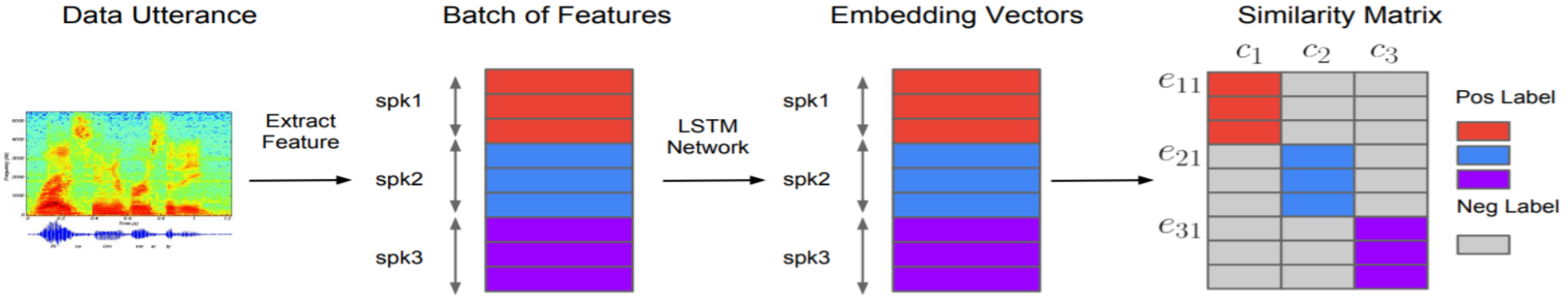
Next, use maybe 50 utterance to test the speaker models

Googles Model

To verity 10 speaker: 1,2,3,4,5,6,7,8,9,10. Each speaker has 20 utterances.
Collect M speakers not includes the first 10.
Each speaker has N utterance. Total utterance N*M
For TE2E, build N*M*M tuples as (arbitrary utterance, random speaker)
For each epoch, should train N*M*M times, each time has N+1 data
The key idea is training a **LSTM/DNN** to output distinguished speaker model.

To verify, random split 20 utterances, say 3 to enroll, 17 to test.
Then it will test 17*10 utterance with 10 speaker model and compute the accept/reject. Further implementation will build speaker model with trained LSTM/DNN, and make decision given evaluation utterance pass through LSTM/DNN.





Instead of construct multiple tuples, GE2E use batch. Each batch contains m speakers, each speaker evaluate n utterance, each speaker enroll n utterance.
to build m*n*m matrix, then update LSTM.

Benefits: Not only far away from one speaker, each speaker will far away from others and close to itself

Sliding Model

- Choose the fricative consonant as identical feature
- Select the position of fricative consonant
- Use mel-filters and MFCC to test if fricative consonant is identical or not. (ultrasound vs normal microphone)
- Compare similarity

Pattern Recognition

- 1. Resize image
- 2. Calculate Perceptual Hash value based on grey value
- 3. Compute hamming distance
- 4. Similarity definition.

SPARTANS WILL.

Result Comparsion

GMM-UBM – Correct Previous Code

- Previously:
- For speaker 1, build GMM[1] using him enroll utterances as his speaker model.
- Build UBM[1] using other 4 speakers enroll utterances.
- Verify: Given an utterance from speaker1, for each sample, do mfcc feature conversion, for each column of mfcc feature, calculate GMM[1] score and UBM[1] score, if the GMM[1].score > UBM[1].score, then correct ++.
- Acc = correct / mfcc_columns

Overall Acc: around 30%.

- Now:
- Collect other 50+ speakers use ultrasound microphone; each speaker collect enough data to build UBM.
- Use another 10 speakers to enroll the system, each one use 3 enroll utterance and adjust UBM to its GMM model for each speaker.

- Verify: use new data contains utterance both other speakers and from 10 speakers, then test the FAR, FRR and EER.

Sliding Window– Correct Previous Code

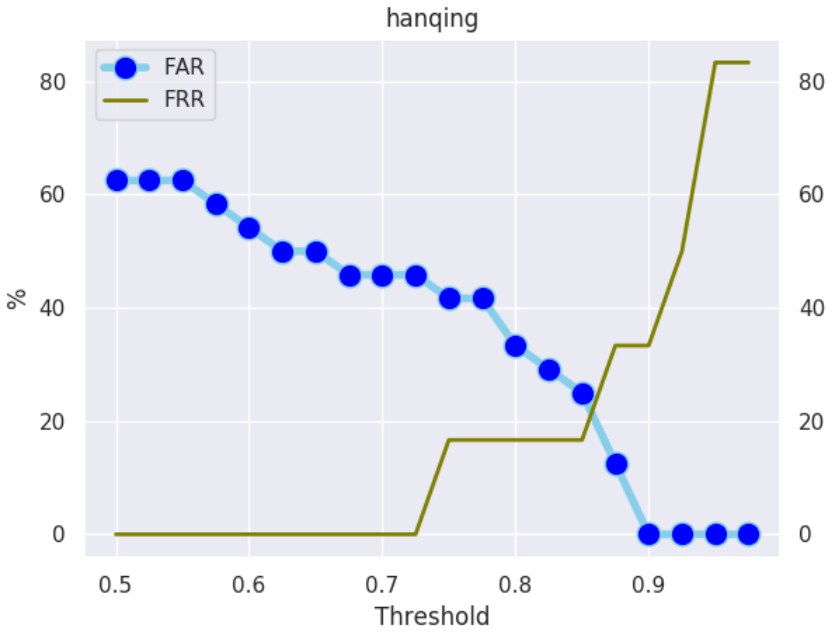
- Previously:
- Hand choose fricative consonant.
- Verify: Roughly set utterance number and set threshold separately to test FRR and FAR.
- Results in better performance.
- Set fix threshold.

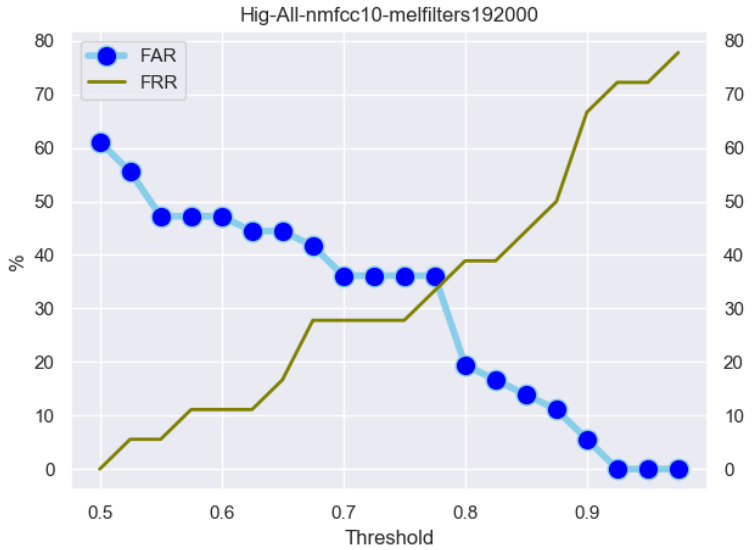
Overall Acc: Wrong procedure leads to wrong data

- Now:
- Choose fricative consonant automatically by using STFT result.
- Slide threshold to test FAR, FRR and EER.

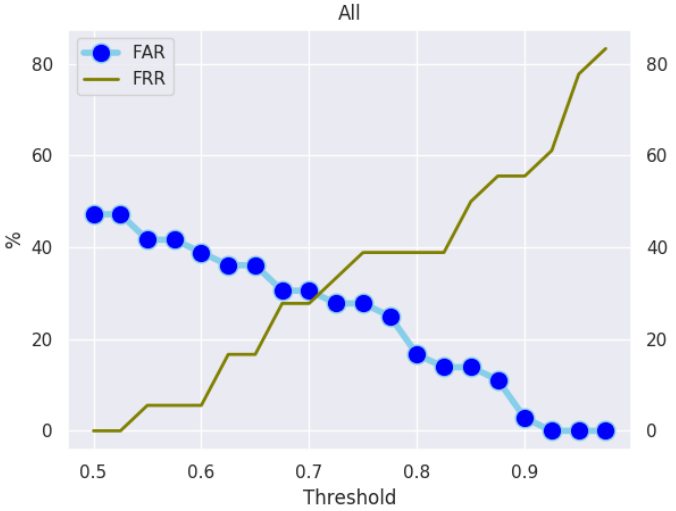
- Shortcoming: error choose fricative position will lead to bad result.
- Current fricative consonant width is only 1.

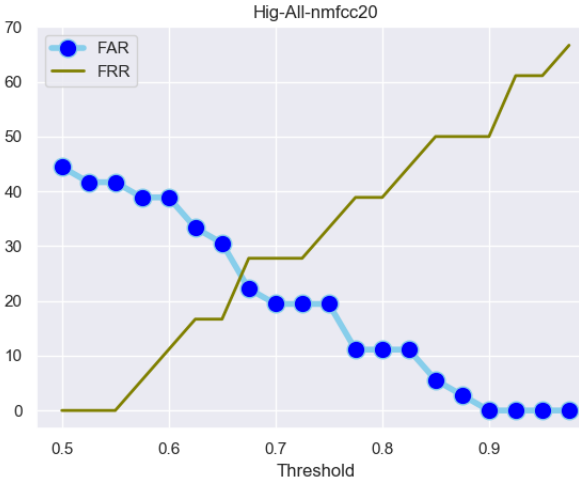
Different mel-filters scale

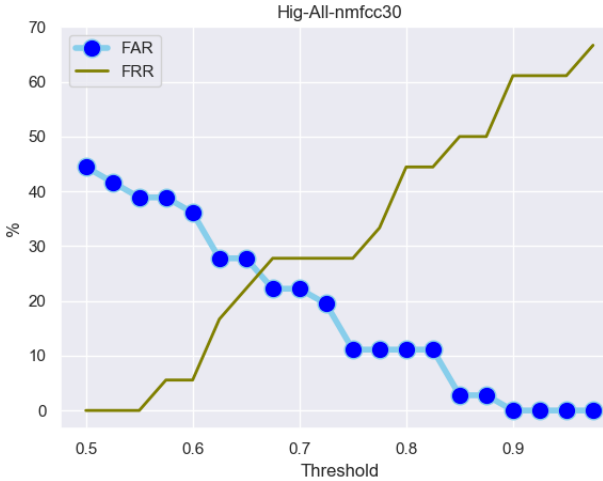




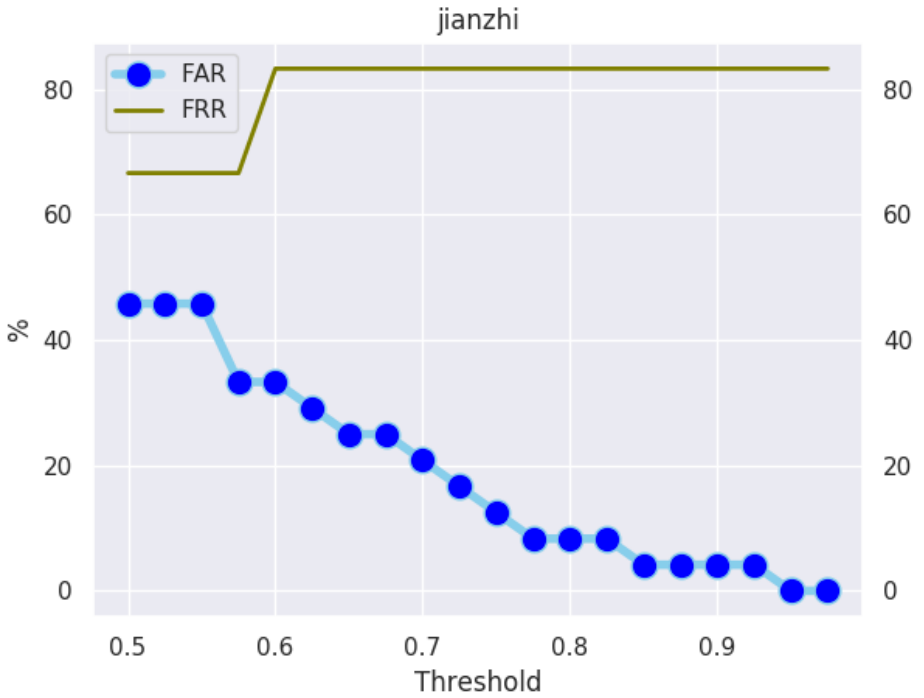
Different nmfccs -> frequency resolution

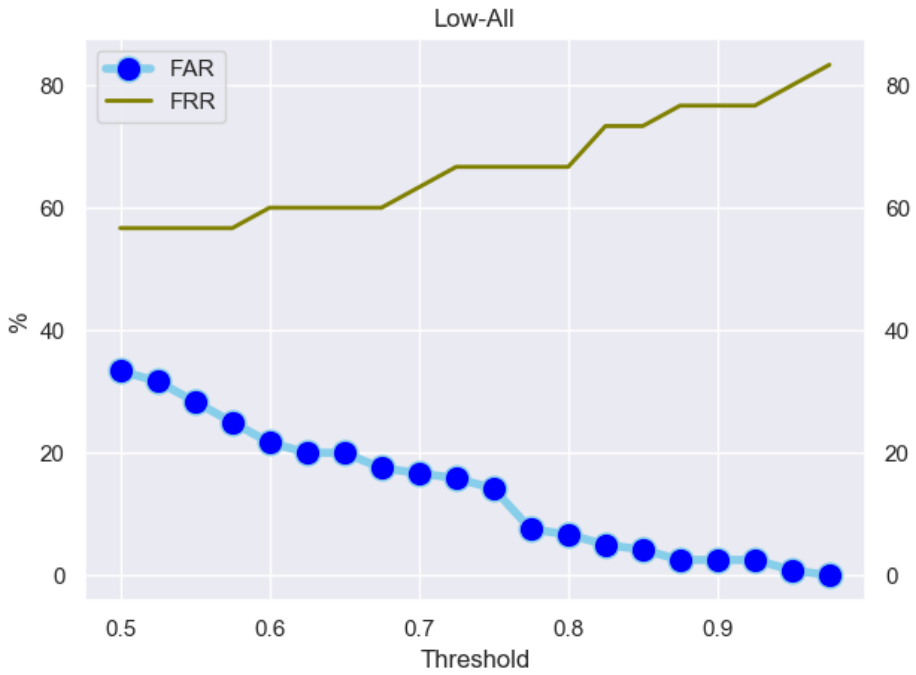






Bad Result- wrong position vs low microphone





What happens using Google Model

Conclusion:

- Sliding Window is useful approach with light weight, no model trained process and performs not bad.
- MFCC features might helps to improve model performance