Carnegie Mellon University

Warm-up for ESPnet Tutorial2 (How to add new tasks/models)

Presented by Jiatong Shi

jiatongs@cs.cmu.edu

Agenda

- Why ESPnet can support different speech tasks? (5min)
- What components we need for a new task in ESPnet (3min)
- Today's task: ASVSpoof (3min)
- Colab going through (1h9min)

Attention!

- Since we follow almost the same installation procedure of ESPnet as Monday's tutorial, we will not go through that part today.
- Please **start the Colab early** and execute the installation procedures ahead of time (you can start to do the clicks during the first explanation period)

- Today, we will modify several lines of source code, which could potentially be lost when disconnected from colab.
- Therefore, please try to **save your modification** in a separate text file so as to avoid losing them.
- We also recommend you going with your own Github account and fork into your space if you have experience in that.

ESPnet in speech research

- **Speech recognition**
- **Speech synthesis**
- **Voice conversion**
- Speaker recognition
- Language recognition
- **Speech emotion recognition**
- **Speaker diarization**
- Speech coding
- Speech perception
- **Speech enhancement**
- **Microphone array processing**
- **Audio event classification and detection**
- **Speech separation**
- **Spoken language understanding**
- Spoken dialogue systems
- **Speech translation**
- **Multimodal processing**
- Speech corpus

Unified form → Unified software design

We design ESPnet by leveraging a **unified** mathematical **form** of **sequence (X) to sequence (Y) transformation f**

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

$$X = (x_1, x_2, \cdots, x_T)$$

$$Y = (y_1, y_2, \cdots, y_N)$$

$$f(\cdot)$$

Speech
Text
English Speech
Noisy Speech

Text
Speech
German Text
Clean Speech

What components we need for a new task in ESPnet

- In short:
  - Task library: the core procedure provided in the task (usually training and inference)
  - Recipe: a recommend stages for the task (usually including data preparation, formatting, preprocess, training, inference, and evaluation as the major stages) → You have already had experiences with the first tutorial

What components we need for a new task in ESPnet

- Task library (What we will focus today)
  - **bin** → core entry of the library. All functions needs to use from here in ESPnet
  - **fileio** → I/O for different kinds of data (e.g., text, sound, rttm, music?)
  - **tasks** → the major step of executing a task
  - **<task_name>** (e.g., asr, tts, st, slu, diar, etc) → task-specific models and their corresponding loss calculation (computational graph construction)

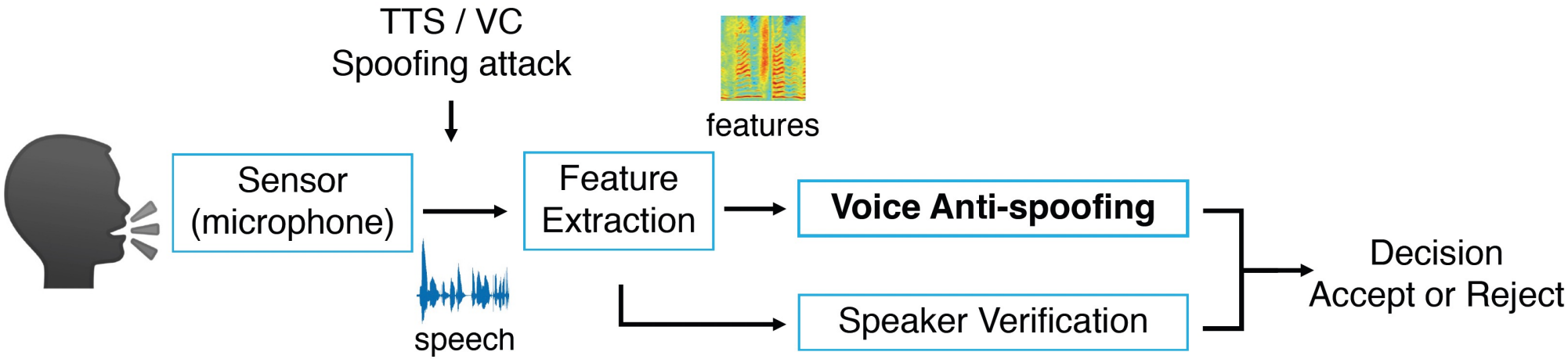
- More in task library that won't be touched today
  - iterators, layers, main_funcs, optimizers, schedulers, samplers, text, torch_utils, train, utils

What components we need for a new task in ESPnet

- Recipe
  - Explicit recipe for a specific corpus
    - We touch that on Monday; will skip it for today (aka. you do not need to worry about this today :-))
  - A template that includes all the recommend stages
    - We prepare 99%, but needs some of your inputs

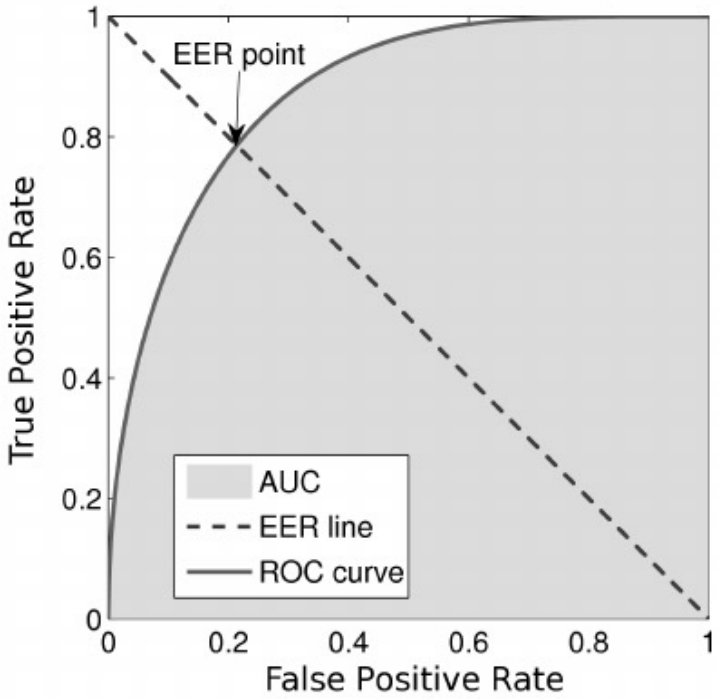
Speaker Verification Anti-Spoofing

- What is the problem:
  - discern spoofing attacks from human natural speech



Wu, Z., Yamagishi, J., Kinnunen, T., Hanilçi, C., Sahidullah, M., Sizov, A., ... & Delgado, H. (2017). ASVspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, *11*(4), 588-604.

Speaker Verification Anti-Spoofing (Evaluation)



- Equal-Error Rate (EER)

https://www.researchgate.net/figure/225180361_fig1_Fig-1-An-example-of-a-ROC-curve-its-AUC-and-its-EER

Time-for-Colab