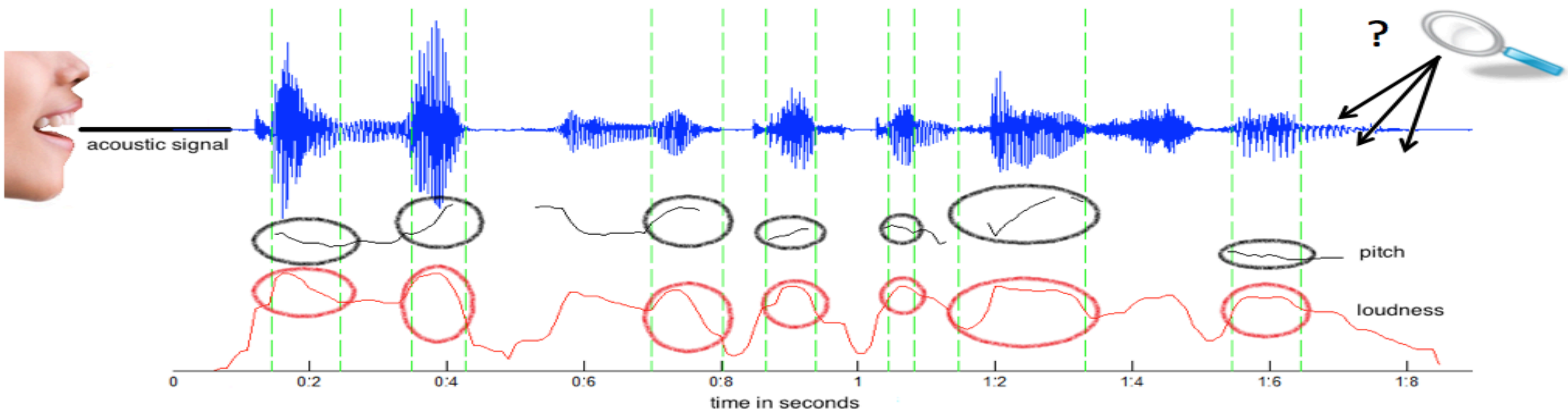
DEEPSER-toolkit

Jaebok Kim
Human Media Interaction
University of Twente

Goals

- Off-the-shelf training models and building speech emotion recognition (SER) applications
- Customizing own models and reproducing experiments
- 100% python codes for soft programmers



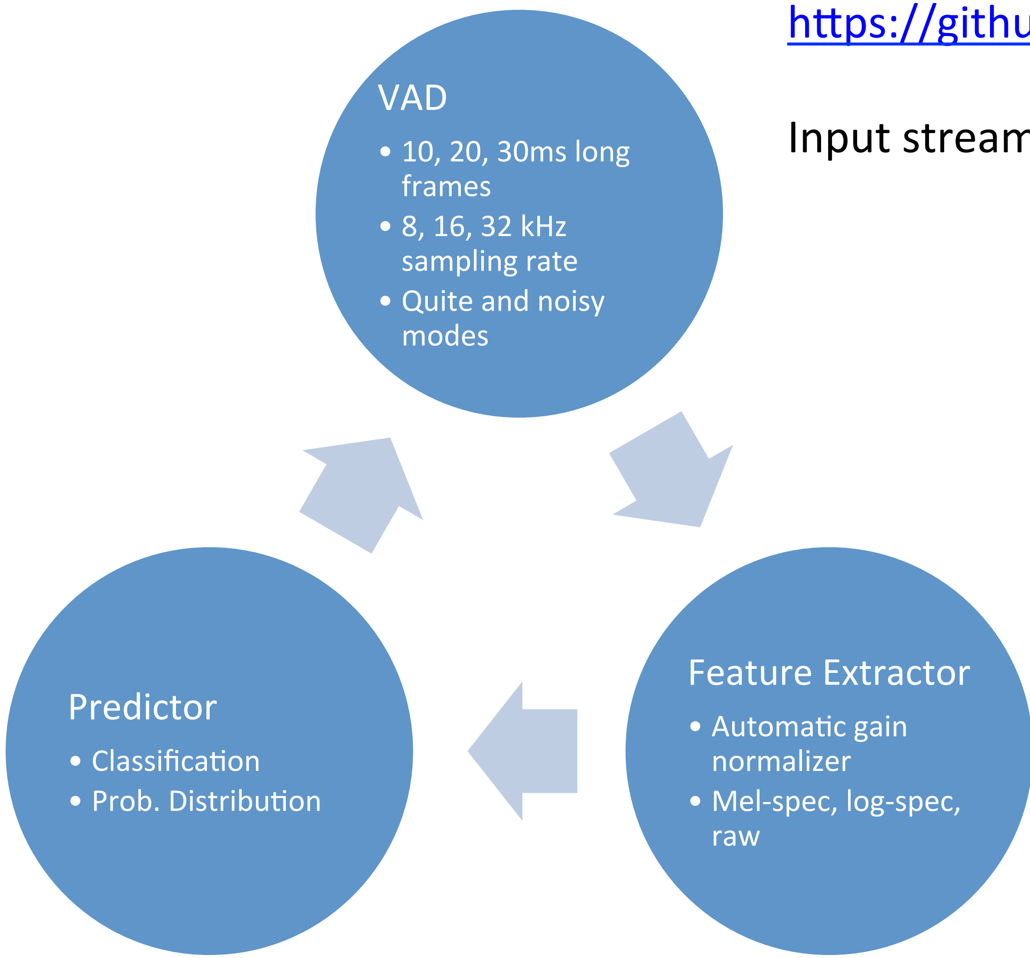
Requirements

- OS
  - MAC OSX >= 10.10.5 (Yosemite)
    - Required packages will be installed by “brew”
  - Ubuntu >= 16.04
    - Required packages will be installed by “apt-get”
  - Windows ? Not tested but may work…
- Python
  - Tested on 2.7x and 3.5x
  - Required packages will be installed by “pip”
- Details of installation can be found in README.md files of each git-hub repository.

Components

- Recognizer (https://github.com/batikim09/LIVE_SER/)
  - Recognizing emotion by using voice activity detection and a trained keras/tensorflow model
- Trainer
  - Extracting features, building and optimizing a keras/tensorflow model

Recognizer



https://github.com/batikim09/LIVE_SER/

Input stream can be both file and live microphone.

An example script (OSX)

- Find your available microphones
  - python ./src/offline_ser.py
- Run recognizer
  - python ./src/offline_ser.py -d_id 1 -p_mode 1 -f_mode 1 -log ./output/live.wav.csv -md ./model/AIBO.si.ENG.cw.raw.2d.res.lstm.gpool.dnn.1.h5 -c_len 1600 -m_t_step 16000 -tasks 'arousal:3,valence:3' –vd 1000 –s_ratio 0.7

-d_id 1: you find your device index is 1

-p_mode 1: classification mode

-f_mode 1: raw wave form, depending on your trained model

-log …: store all recognized results into a file

-md …: your trained model

-c_len 1600: time-steps in each contextual window

-m_t_step 16000: maximum time-steps for each utterance

-tasks …: your classification tasks and their number of classes

-vd 1000: minimum duration of speech to recognize

-s_ratio 0.7: minimum proportion of speech segments in total segments

VAD and feature extraction

Is the total segments longer than 1000 ms?
Are 70% of total segments speech segments?

If both rules are satisfied, features are extracted.

The time steps in a contextual window and maximum time steps per utterance depend on features & model.

For example, log spectrogram windows have 10 time steps of each 25ms long frame but overlaps, 10 windows per utterance make maximum time steps 100 (~= 1sec).
Raw form windows have 1600 time steps (1600 samples ~= 100ms), 10 windows per utterance make maximum time steps 16000 (=1sec).

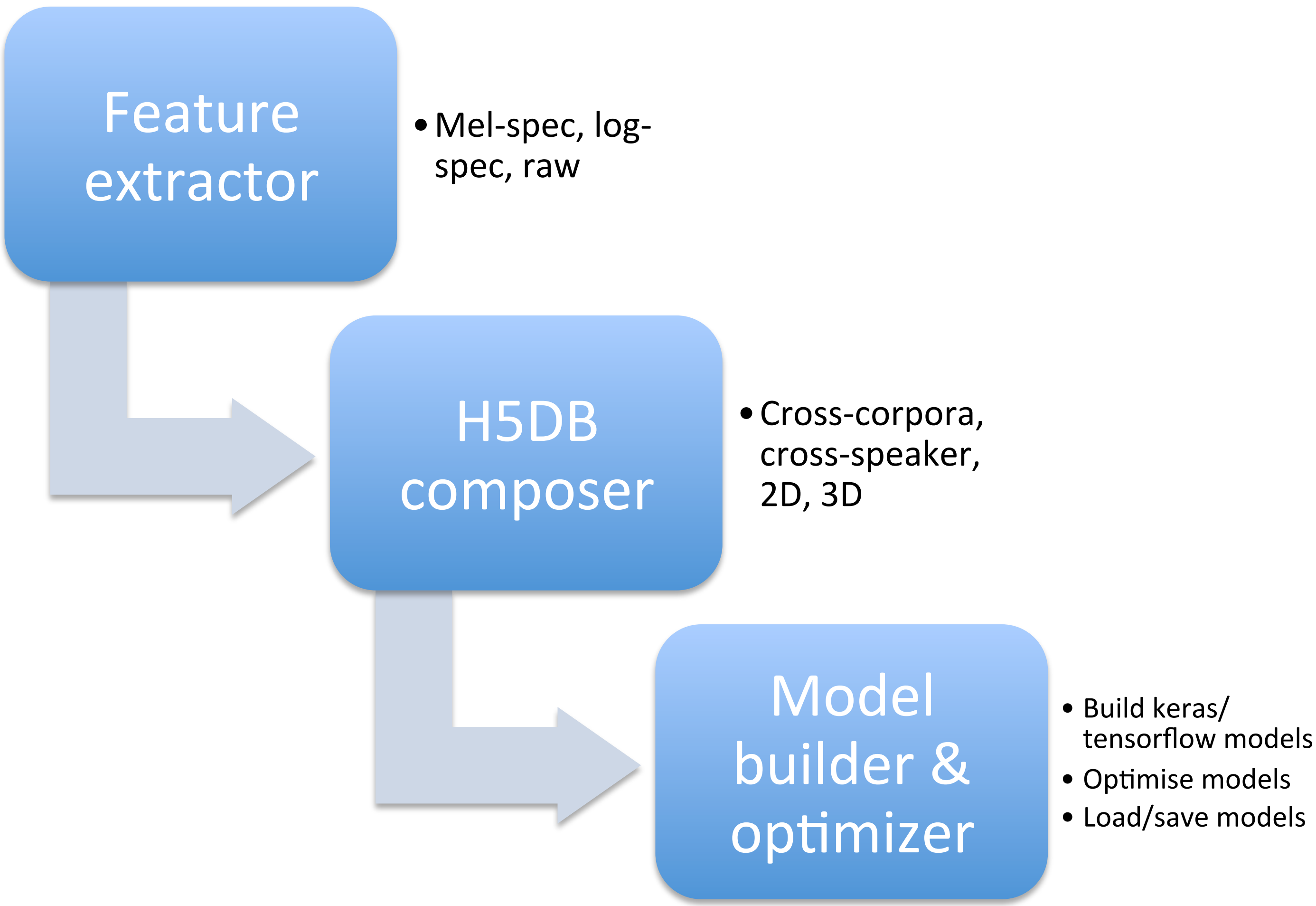
Pre-trained models

- Two English models provide arousal and valence (3 class each) predictions
  - ./model/si.ENG.cw.raw.2d.res.lstm.gpool.dnn.1.h5
    - Raw, RESNET-LSTM-DNN
    - Based on “Deep Temporal Models using Identity Skip-Connections for Speech Emotion Recognition, ACMMM17”
  - ./model/si.ENG.cw.mspec_mm.3d.rc3d.1.h5
    - Mel-spec + RESNET-3DCNN
    - Based on “Learning spectro-temporal features with 3D CNNs for speech emotion recognition, ACII17”
- Their performances are more less same.

Automatic Gain Normalization

- Gains are really crucial to performance.
- After starting the script, practice several utterances with various gains.
- It collects gains and finds minimum and maximum gains for min-max normalization.

Trainer



Feature extractor
• Mel-spec, log-spec, raw
H5DB composer
• Cross-corpora, cross-speaker, 2D, 3D
Model builder & optimizer
• Build keras/ tensorflow models
• Optimise models
• Load/save models

Meta file (e.g. ./meta/sanity.enterface.txt)

- A meta file is required to extract features and build a h5 DB.

n_train_data.name	class	sid	corpus_id	gender	acted	arousal	valence
./wav/enterface/ENT_s01/s1_an_1_R.wav	5	0	2	0	1	2	0
./wav/enterface/ENT_s01/s1_an_2_R.wav	5	0	2	0	1	2	0
./wav/enterface/ENT_s01/s1_an_3_R.wav	5	0	2	0	1	2	0
./wav/enterface/ENT_s01/s1_an_4_R.wav	5	0	2	0	1	2	0
./wav/enterface/ENT_s01/s1_an_5_R.wav	5	0	2	0	1	2	0

- The first column shows wave files that we extract features from.
- Other columns have codes or labels that represent meta information and can be used for classification tasks.
- Meta files are tab-separated text files.

Feature extractor

- After you run the following script:
  - python ./src/extract_feat_temporal_LLD_rosa.py -f ./feat/RAW/ -m ./meta/sanity.enterface.txt --gain_stat
- You get gain information: min and max.
- You need to put these for gain normalization.
  - python ./src/extract_feat_temporal_LLD_rosa.py -f ./feat/MSPEC/ -m ./meta/sanity.enterface.txt -min -1.0541904 -max 1.1097699
- We have several options to extract various types of features. See details by
  - python ./src/extract_feat_temporal_LLD_rosa.py –h
  - Currently, Mel-spec, log-spec, PCA whitened log-spec, raw wave forms are supported.
- This script generates another meta file ("sanity.enterface.txt.mspec.out") that includes a new column showing feature files and will be used to build a h5 DB.

Your own feature sets?

- You can also use own features.
- For example, by using Opensmile, extract feature vectors for each utterance separately, do not append them to a single file.
- Prepare a metafile that tell which feature vector files have which meta information.
- Time-invariant feature sets that use statistical functionals are not supported now. Our tools support only a time series of feature vectors.

Building a h5 DB

- For faster training, we use a h5 DB. To build such a DB, we run a script:
  - python ./src/h5db_builder.py -input ./meta/sanity.enterface.txt.mspec.out -m_steps 100 -c_idx 2 -n_cc 43 -c_len 10 --two_d -mt 1:2:4:5:6:7 -out ./h5db/ENT.MSPEC.2d.3cls.av
  - We need to specify which column in the meta file indicates an index of cross-validation. For example, "2"th column shows speaker id, so we use it for speaker-independent cross-validation by specifying "-c_idx 2". The total number of folds (speakers) is "43" too.
  - Also, we can pass many indices of meta information. Some of them can be used as labels.

Aggregated corpora

- If you have multiple corpora and build a speaker-independent cross-validations, run the following script:
  - python ./src/h5db_builder_cc_sid.py -input ./meta/sanity.aibo_enterface.txt.mspec.out -m_steps 100 -c_ids 0,1 -c_idx 3 -s_idx 2 -c_len 10 --two_d -mt 1:2:4:5:6:7 -out ./h5db/AE.RAW.2d.3cls.av
  - c_idx specifies corpus id and s_idx does speaker id. Since there are overlaps of speaker ids between corpora, it automatically generates new speaker ids.
  - You can choose corpora among aggregated corpora by specifying c_ids too.

A H5 DB structure

- If we set cross-validation options (-n_cc, -c_idx), the DB built has 4 data sets as follows:
  - feat: containing a feature matrix
  - label: meta label vectors
  - start_indice: start index of each fold
  - end_indice: end index of each fold
- If you do not specify them, it will have only feat and label.

A feature matrix

- For 2D layers (e.g. 2D CNN)
  - #samples x #time-steps x #channel(always, 1) x #context-window-length x #feature-dimensions
- For 3D layers (e.g. 3D CNN)
  - #samples x #channel(always, 1) x #time-steps x #context-window-length x #feature-dimensions
- These will be unified in soon.

Model build and optimization

- You can choose many options and configure your own networks.
- See examples of scripts.
  - basic.sh
    - From simple networks (DNN, LSTM) to complex networks (CNN-LSTM, RESNET, 3DCNN)
  - pretrain.sh
    - Save and load models, finetunning
  - balance_learning.sh
    - Balanced learning, prediction, and re-sampling

Next steps

- Nice demos, visualization? Students' projects?
- Provide more models recognizing various contexts: gender…