Deep Learning for Intelligent Signal Processing

Tuesday

Deep Learning Summer School 2019

Prof. mult. Dr. habil.

Björn W. Schuller

MD, CSO, EiC, FIEEE

Imperial College London

Universität Augsburg University

audEERING

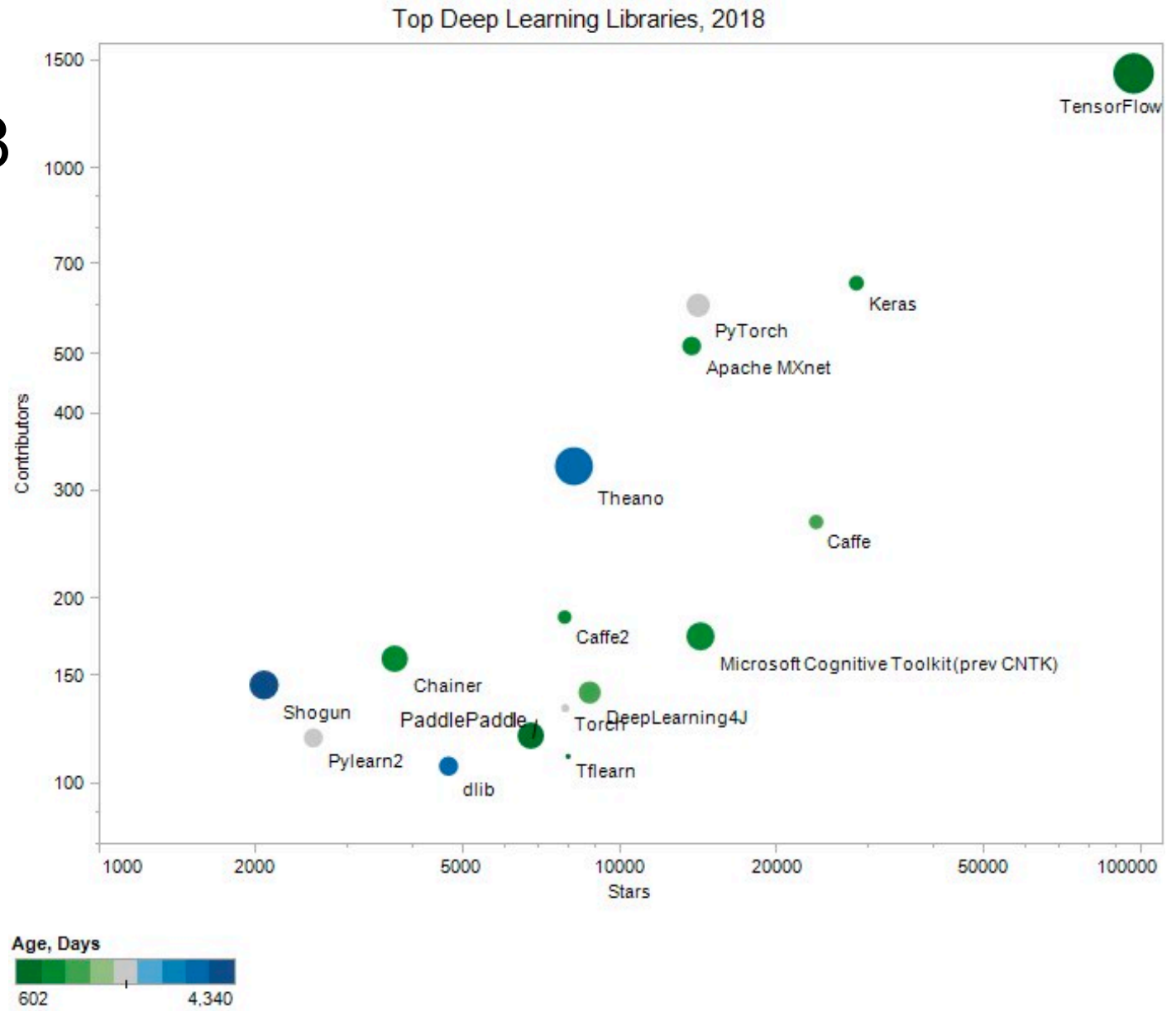
intelligent Audio Engineering

DL Libs

Universität Augsburg University

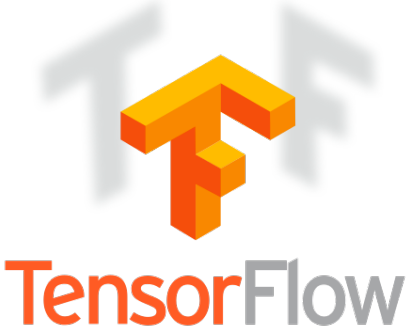
- Top os DL Libs 2018
- Source: KDnuggets

- Github stars and contributors
- log scale for both axes
- Colour: age in days
  (greener - younger, bluer - older)



DL Libs

- TensorFlow 2.0 Beta (Google)
- https://www.tensorflow.org/



- Python, JAVA
- Keras (also Theanos, Microsoft Cognitive Toolkit)

- TensorFlow Lite
- TPU, ML Crash Course

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)

model.evaluate(x_test, y_test)
```

DL Libs

Universität Augsburg University

1. **TensorFlow** Google Brain Team ML, easy to transition from research prototype to production system.
2. **Keras** ihigh-level NN API, written in Python, capable of running on top of TensorFlow, CNTK, or Theano.
3. **Caffe** DL framework optimisised for expression, speed, modularity
4. **Microsoft Cognitive Toolkit (Previously CNTK)** unified DL toolkit, describes NNs as series of computational steps via directed graph
5. **PyTorch** Tensors and Dynamic NNs in Python, strong GPU acceleration
6. **Apache MXnet** DL framework for efficiency, flexibility. mix symbolic and imperative programming to maximise efficiency and productivity
7. **DeepLearning4J** distributed NN lib in Java and Scala
8. **Theano** allows to define, optimise, evaluate math expressions w/ multi-dimensional arrays efficiently. developments cease
9. **TFLearn** modular and transparent DL lib on top of TensorFlow to facilitate and speed-up experiments,
10. **Torch** provides utilities for accessing files, serializing objects of arbitrary types

DL Libs

11. **Caffe2** lightweight, modular, and scalable DL framework. Caffe2: expression, speed, and modularity
12. **PaddlePaddle** (PArallel Distributed Deep LEarning) easy-to-use, efficient, flexible and scalable DL platform by Baidu
13. **DLib** modern C++ ML toolkit to solve complex real world problems
14. **Chainer** Python-based, standalone DL models. flexible, intuitive, and high performance , full range of DL models
15. **Neon** Python-based DL lib. Ease of use, highest performance
16. **Lasagne** lightweight lib to build and train NNs in Theano

Contents

Universität Augsburg University

- CAS^{2}T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- openXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

Contents

Universität Augsburg University

- CAS^{2}T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- openXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

CAS²T – Efficient Data Collection

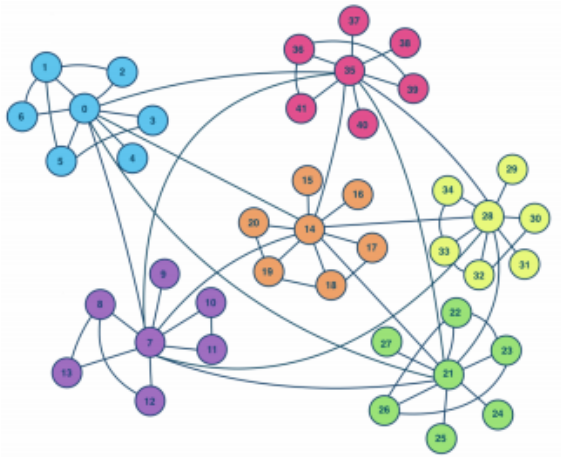
Universität Augsburg University

- Contemporary deep learning topologies
  - Require considerably more training data than conventional machine learning approaches
    - E.g., end-to-end learning, reinforcement learning
- Cost of collecting such data is large:
  - Time and labour
    - Particularly true in affective and behavioural computing
- Suitable training material is already freely available
  - Need an efficient method to harvest social media resources

Universität Augsburg University

Cost-efficient Audio-visual Acquisition via Social-media Small-world Targeting (CAS²T)

- Purpose-built for data collection from social media
  - Fast, Efficient, Reliable
  - Minimal manual effort
- Utilises a unique combination of
  - Complex Systems Theory
  - Unsupervised Audio Analysis
  - Semi-Supervised Active Learning
  - Crowdsourcing



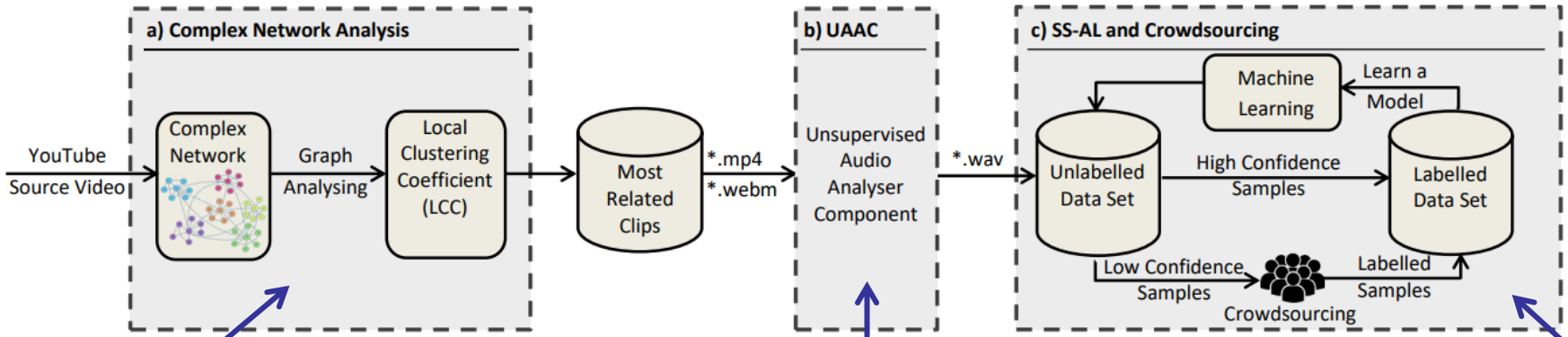
openCoSy

S. Amiriparian, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, “CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms,” in *Proceedings of the 7th biannual Conference on Affective Computing and Intelligent Interaction, ACII 2017*, (San Antonio, TX), IEEE, Oct. 2017. pp. 340-345

CAS²T – Efficient Data Collection

Universität Augsburg University

- CAS²T consists of three core components
  - Complex Network Analyser, Unsupervised Audio Analyser, Intelligent Annotations



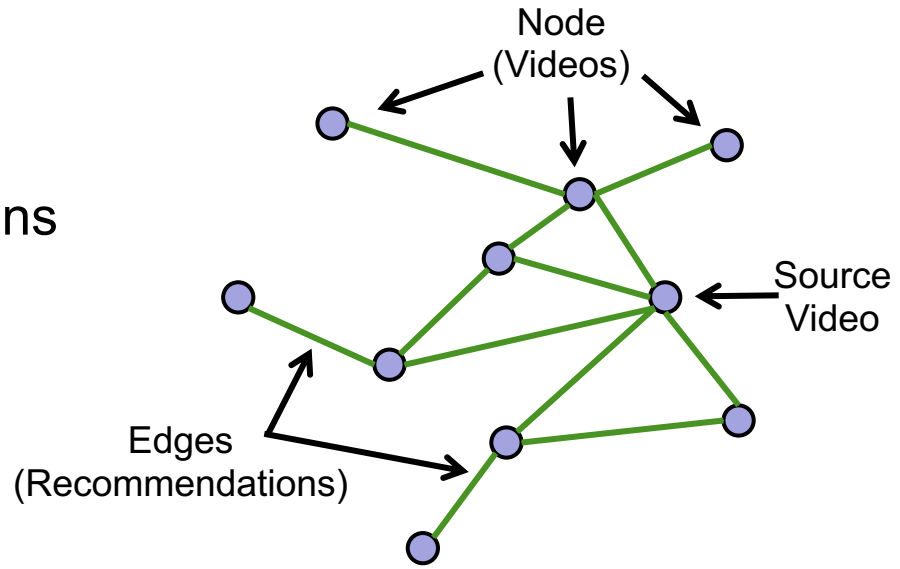
Complex systems analysis is used to identify an initial set of `related' multimedia clips available on social media

Unsupervised Audio Analyser Component performs event detection to isolate audio instances of potential interest

Semi-supervised active learning (SS-AL) and crowdsourcing is used to label the data with minimal human intervention

Complex Network Analyser

- YouTube has large recommendation system
  - Based on number of views, titles, descriptions, viewer rankings…
- Underlying assumption made in CAS²T
  - The content of videos by the YouTube recommender is similar
- Therefore,
  - Starting with a source video
    - Build an *undirected graph* of recommendations
      - **Vertices** - related content
      - **Edges** - number of recommendations



Complex Network Analyser

- Resulting graph is a mapping of YouTube's recommendation space in relation to the source video
  - The *Local Clustering Coefficient* is used to identify related videos

$$C_{v_i} = \frac{2n}{k_{v_i}(k_{v_i} - 1)}$$

    - C_{v_i} is the LCC for vertex vi
    - n is the number of edges associated with vertex v_i
    - k_{v_i} is the number if neighbours associated with vertex v_i
  - Videos with a high number of edges (i.e. recommendations), will have a high LCC value

Software is publicly available

- Available at:
  - https://gitlab.com/openCoSy/CAS2T.git
- Dependencies:
  - JAVA
    - https://java.com/en/download/
  - YouTube API
    - https://developers.google.com/youtube/v3/getting-started

CAS²T – Efficient Data Collection

Universität Augsburg University

Key steps:

1. Insert valid YouTube API key
2. Choose new database
3. Choose openSMILE config file
4. Write first video url or id
5. Select videos to iterate
6. Download and convert videos
7. Audio Event Detection (optional)
8. Extraction of LLDs
9. Correlation Coefficient

openCoSy

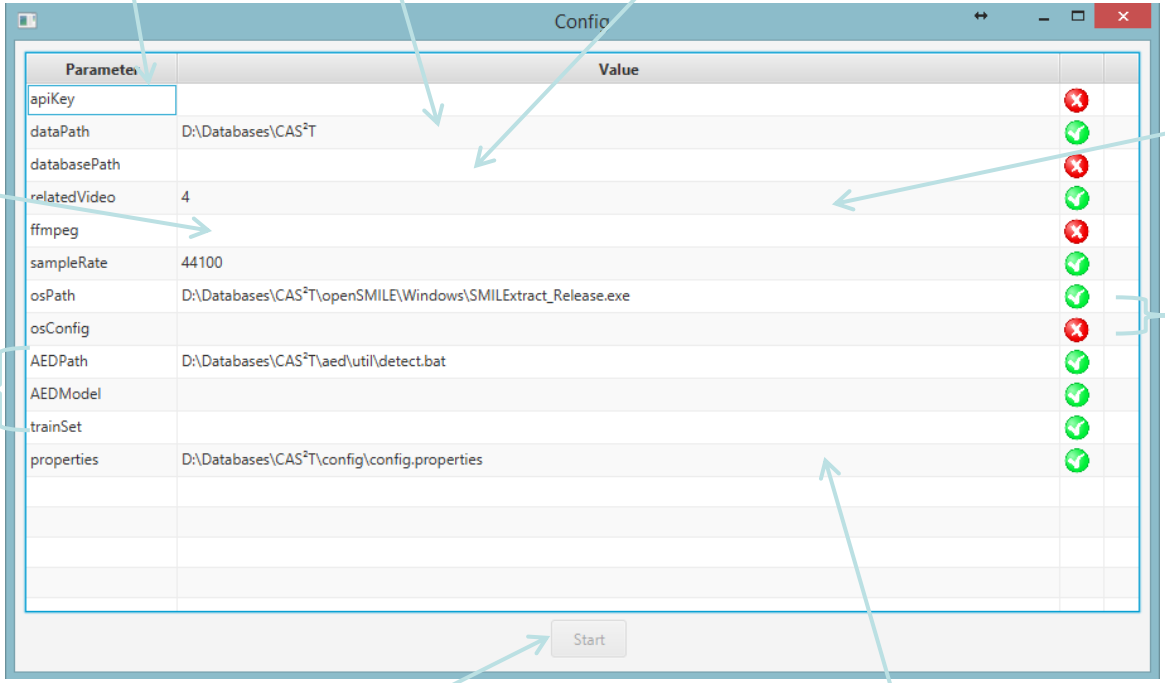
CAS²T – Efficient Data Collection

Universität Augsburg University

API KEY from Google Developers

Folder for downloads

Database (.db) to store video information



Path to Audio Converter

Size of Search Space

For Audio Event Detection

Path to openSMILE and specific config

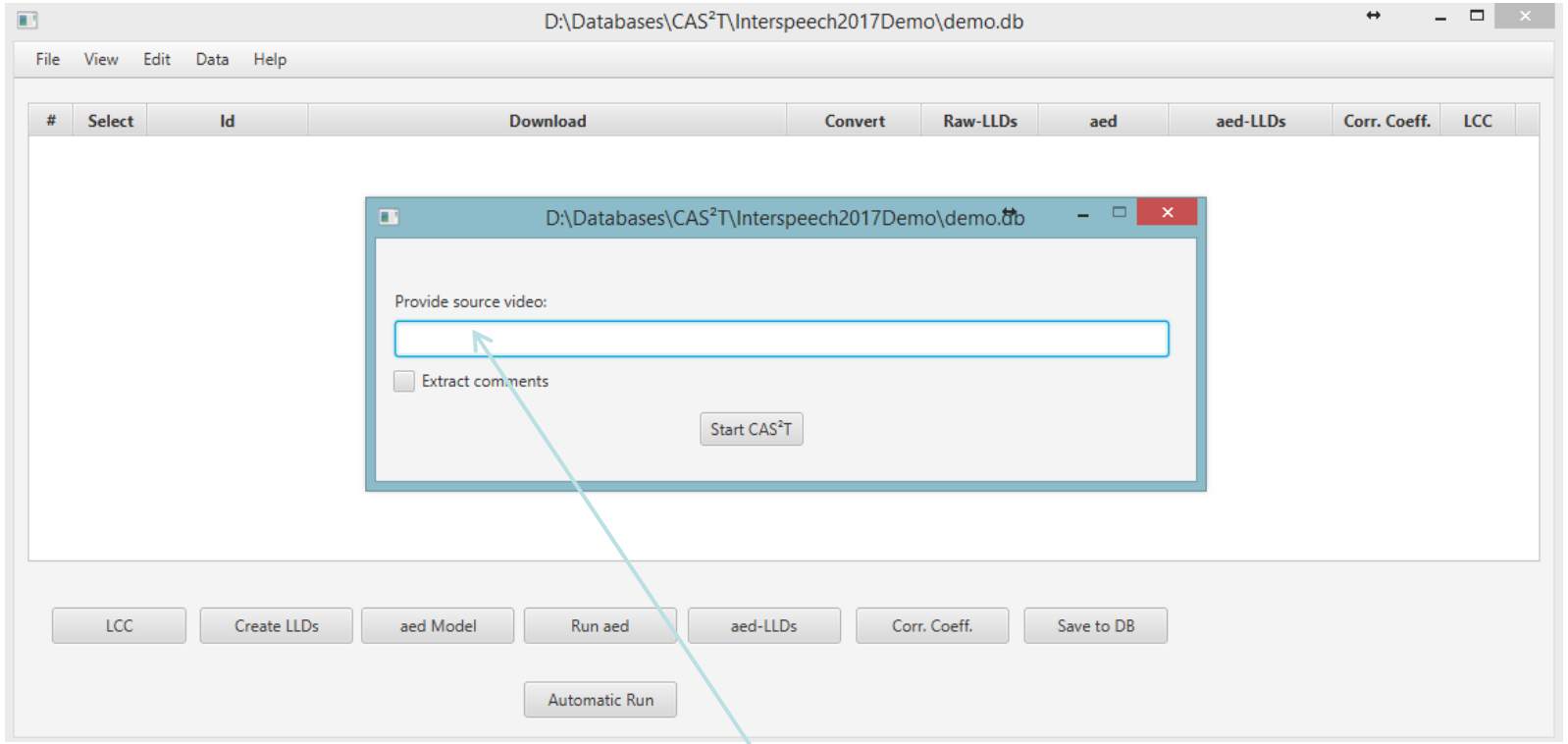
Start Searching

To save these properties

openCoSy

CAS²T – Efficient Data Collection

Universität Augsburg University



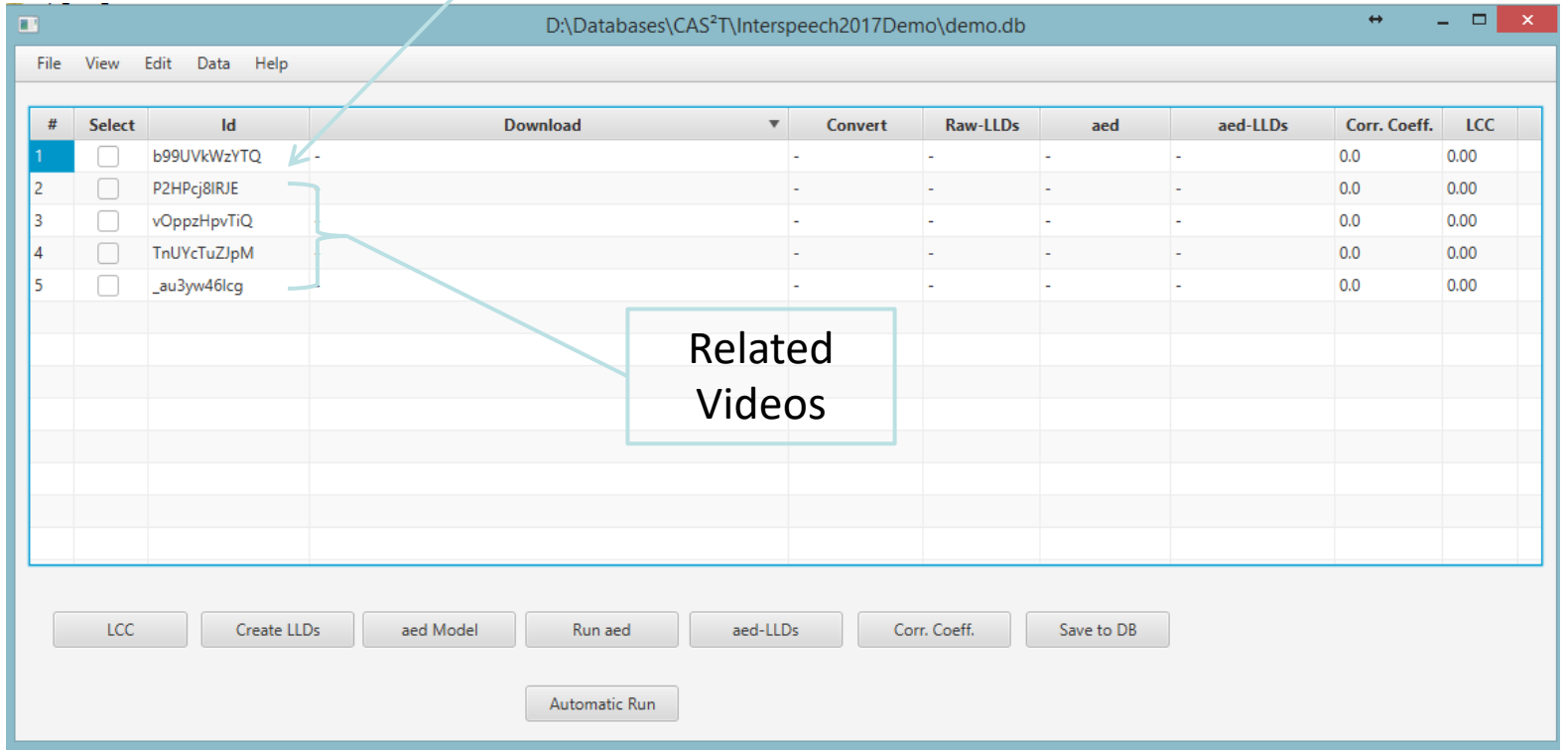
Link to Initial YouTube Video

openCoSy

CAS²T – Efficient Data Collection

Universität Augsburg University

Initial YouTube Video



Related Videos

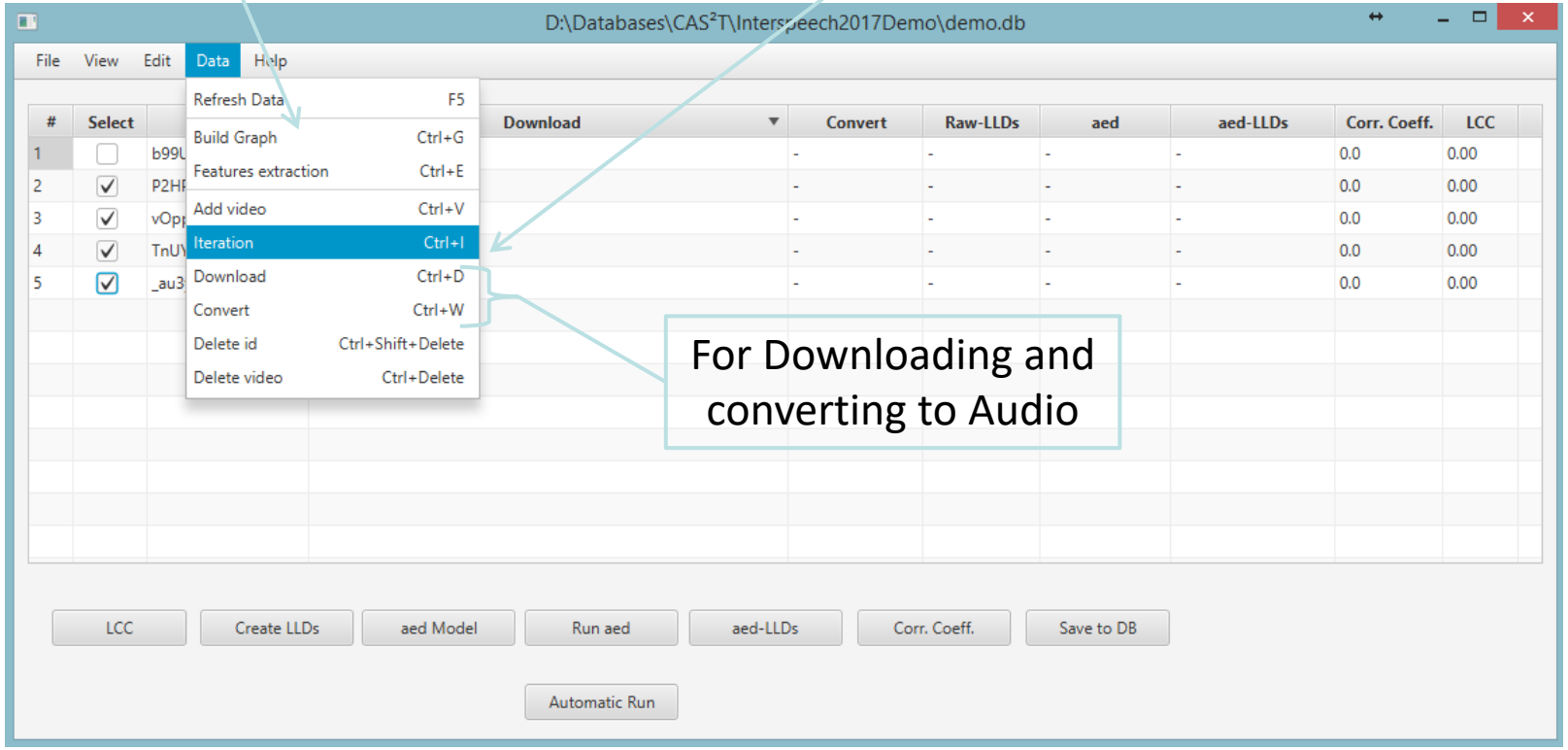
openCoSy

CAS²T – Efficient Data Collection

Universität Augsburg University

Visualise Constructed Graph

Select Interesting Videos and re-iterate



For Downloading and converting to Audio

openCoSy

24 Hour Data collection challenge

- Data collected using a mix of CAS²T and crowdsourcing
- All data collected and labelled over 24 hours

	Train				*Evaluation*			
Tasks	l_{total}	l_{min}/l_{max}	σ	n_{tgt}/n_{sp}	l_{total}	l_{min}/l_{max}	σ	n_{tg}/n_{sp}
Freezing	75.9 m	2.0 s/29.4 s	5.8 s	409/205	22.4 m	2.0 s/28.6 s	5.9 s	91/80
Intoxication	139.7 m	2.0 s/29.9 s	6.5 s	514/555	16.7 m	2.0 s/24.8 s	5.3 s	97/55
Screaming	53.6 m	2.0 s/29.9 s	7.6 s	203/172	22.0 m	2.1 s/29.9 s	5.5 s	111/78
Threatening	106.6 m	2.0 s/29.8 s	7.4 s	559/93	45.8 m	2.0 s/29.2 s	5.2 s	163/278
Coughing	94.3 m	0.5 s/28.8 s	3.5 s	1 553/535	63.9 m	0.5 s/23.2 s	2.7 s	1 080/491
Sneezing	6.7 m	0.5 s/8.0 s	1.3 s	114/124	9.2 m	0.5 s/9.3 s	1.4 s	150/141

openCoSy

S. Amiriparian, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, "CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms," in *Proceedings of the 7th biannual Conference on Affective Computing and Intelligent Interaction, ACII 2017*, (San Antonio, TX), IEEE, Oct. 2017. 6 pages

24 Hour Data collection challenge

		SVM		*BoAW+SVM*				*CNN*
			Evaluation				**Evaluation**	**Evaluation**
Tasks	***Feature Set***	*C*	*UA (WA)*	*C*	*cw*	*S*	*UA (WA)*	*UA (WA)*
Freezing	*IS09-emotion*	10^{-6}	**70.19** (70.76)	10^{-6}	30	4 200	67.48 (69.01)	56.91 (58.48)
	MFCCs	–	–	10^{-6}	20	3 600	65.56 (66.08)	51.06 (53.22)
Intoxication	*IS09-emotion*	10^{0}	64.71 (62.50)	10^{-2}	20	3 600	**72.57** (73.03)	66.84 (69.74)
	MFCCs	–	–	10^{0}	20	3 600	66.72 (69.08)	67.54 (67.11)
Screaming	*IS09-emotion*	10^{-3}	89.21 (88.88)	10^{-5}	30	4 200	**96.98** (97.35)	89.22 (90.45)
	MFCCs	–	–	10^{0}	20	3 600	93.97 (94.71)	87.30 (88.89)
Threatening	*IS09-emotion*	10^{-5}	**73.82** (72.10)	10^{-2}	30	4 200	66.26 (72.33)	71.85 (70.75)
	MFCCs	–	–	10^{-2}	30	4 200	66.96 (72.11)	70.30 (68.41)
Coughing	*IS09-emotion*	10^{-3}	95.36 (96.37)	10^{-4}	30	4 200	96.69 (96.05)	95.44 (94.72)
	MFCCs	–	–	10^{-4}	30	4 200	**97.58** (97.52)	93.60 (92.04)
Sneezing	*IS09-emotion*	10^{-3}	79.26 (79.38)	10^{-4}	20	3 600	76.44 (76.63)	**85.16 (85.22)**
	MFCCs	–	–	10^{-3}	20	3 600	79.83 (78.46)	80.17 (80.41)

openCoSy

S. Amiriparian, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, "CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms," in *Proceedings of the 7th biannual Conference on Affective Computing and Intelligent Interaction, ACII 2017*, (San Antonio, TX), IEEE, Oct. 2017. 6 pages

Contents

Universität Augsburg University

- CAS²T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- openXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

- **Intelligent gamified annotation solutions**
  - The process of gathering annotated data is expensive and time-consuming
  - Making use of crowdsourcing is a viable alternative to reduce these costs
  - Gamified elements encourage people to voluntarily sign up for annotations tasks
  - Intelligent quality control solutions used to ensure the obtained labels are off high quality

iHEARu-PLAY – Intelligent Data Annotation

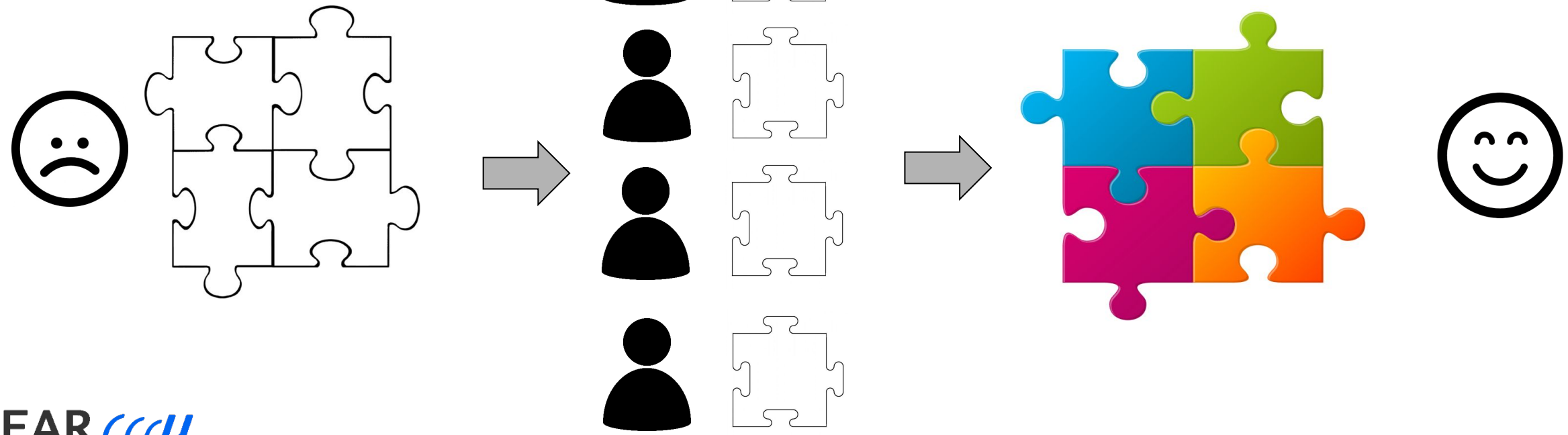
Universität Augsburg University

Crowdsourcing:

Large projects broken into 'microjobs'..

..processed at the same time by multiple users..

.. results then merged…



iHEARu PLAY

Universität Augsburg University

iHEARu-PLAY: A Web-based platform for intelligent, gamified & crowdsourced annotations

- Multi-label, holistic annotation of audio-visual data
- Gamified, fun, and playful environment
- Inbuilt Intelligence and Quality Management
- Key Features
  - Annotation Mode
  - Recording Mode
  - Voice Analysis Mode
  - Researcher Portal

iHEARu PLAY

iHEARu-PLAY – Intelligent Data Annotation

- Available:
  - https://www.ihearu-play.eu/
- Dependencies
  - An internet connection
  - Username and Password

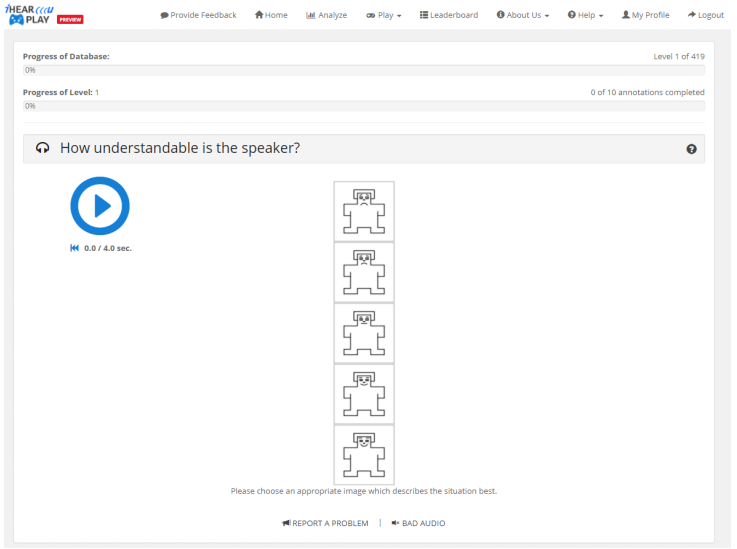
S. Hantke, F. Eyben, T. Appel, and B. Schuller, “iHEARu-PLAY: Introducing a game for crowdsourced data collection for affective computing,” in *Proc. 1st International Workshop on Automatic Sentiment Analysis in the Wild (WASA 2015) held in conjunction with the 6th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2015)*, (Xi’an, P. R. China), pp. 891–897, AAAC, IEEE, September 2015.

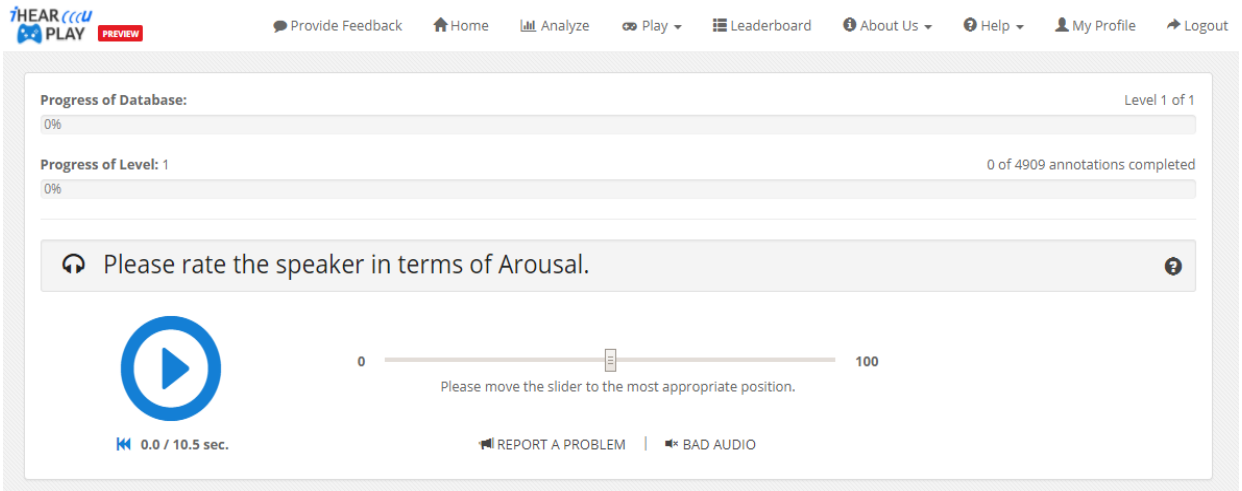
iHEARu-PLAY – Intelligent Data Annotation

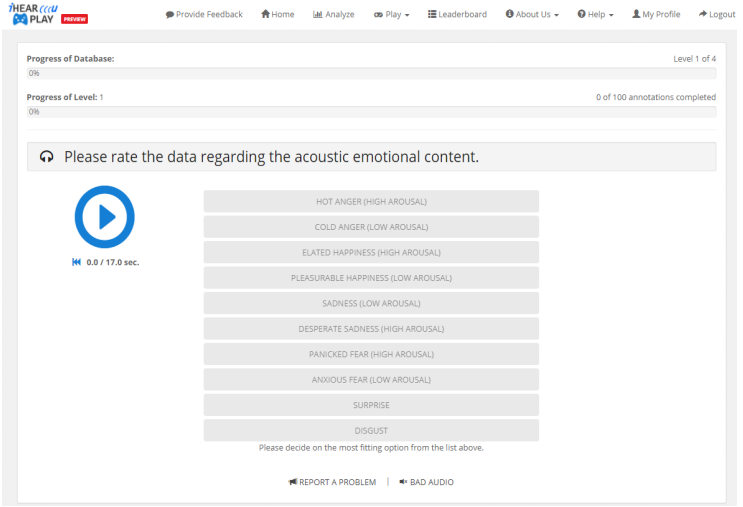
Universität Augsburg University

Annotation Mode

- Discrete: *single, multi, numeric*
- Continuous: *1D, 2D, time dependent*
- Self Assessment
- Pairwise Comparisons





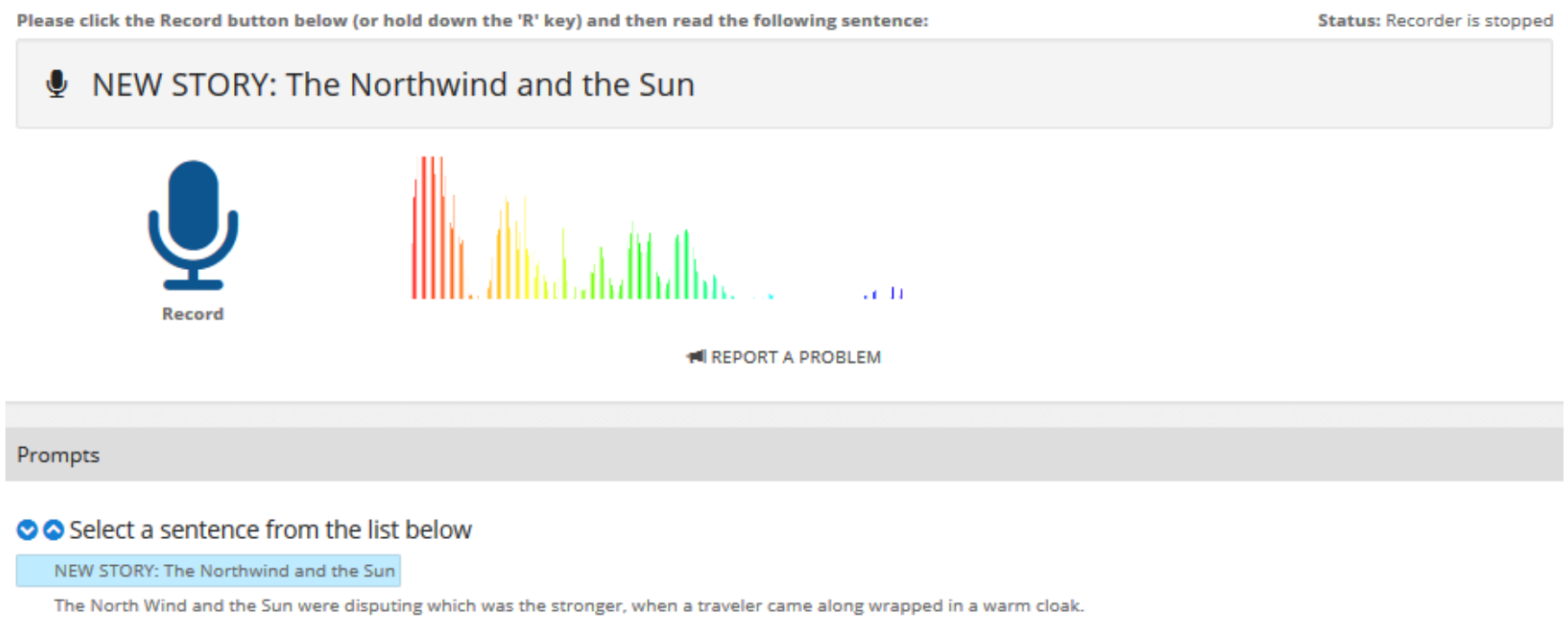


iHEARu PLAY

Universität Augsburg University

Recoding Mode

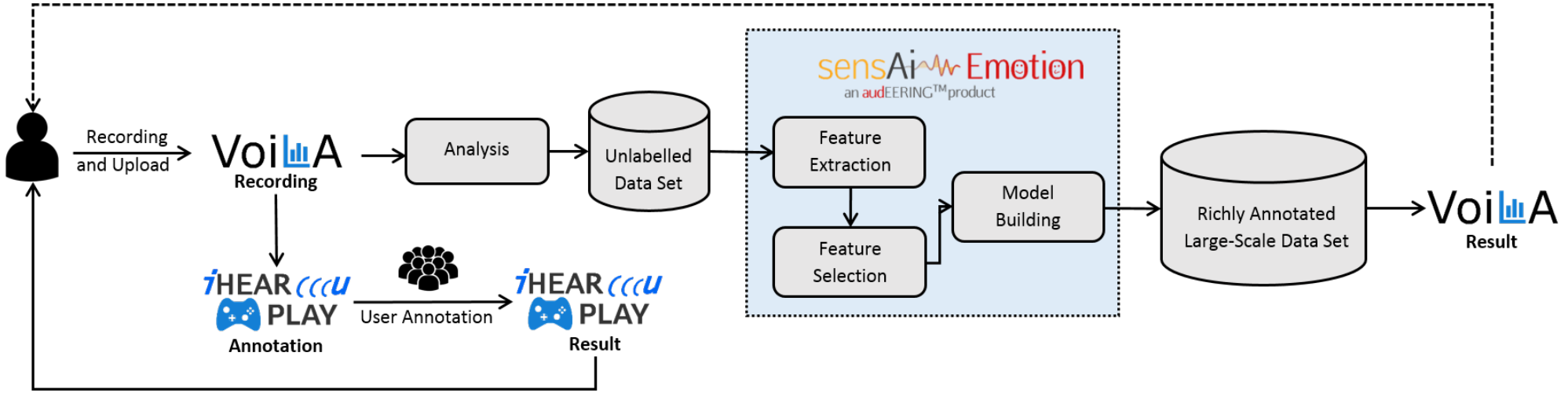
- Records a specific prompt with a distinct state or trait, e.g. happy, sad, whisper, ….



iHEAR PLAY

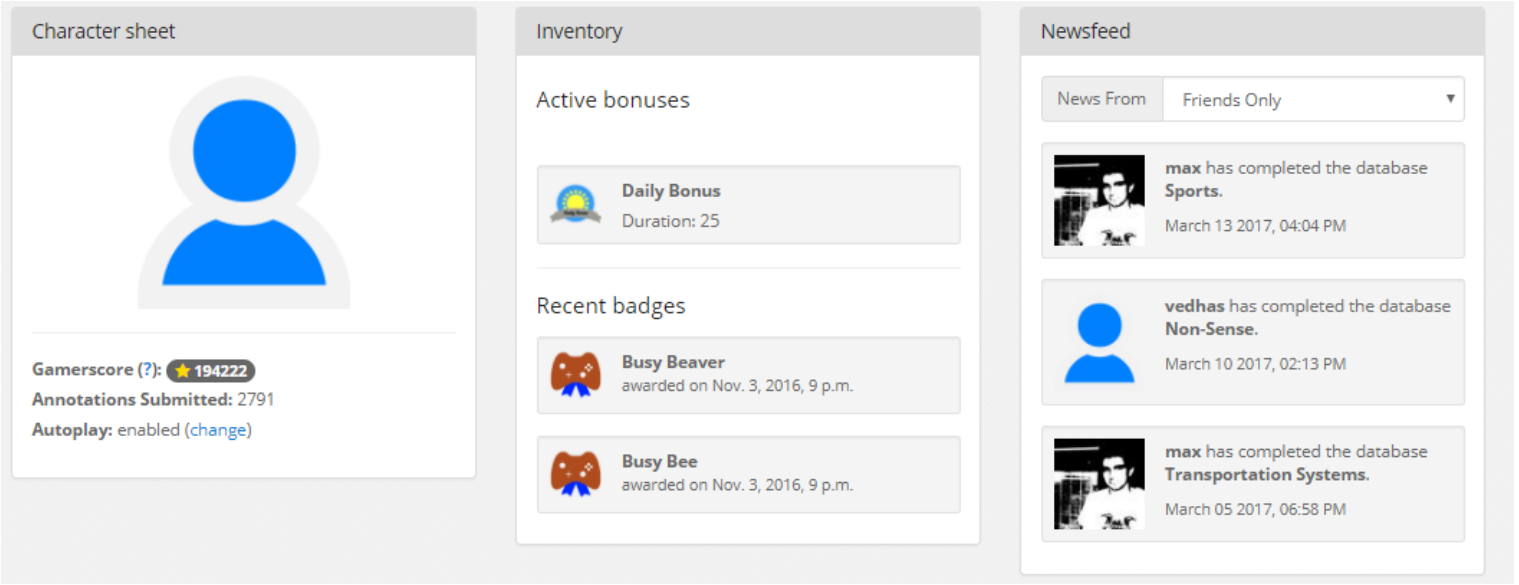
Voice Analysis Mode (VoiLA)

- Emotion, gender, interest level
- sensAI webservice (from audEERING)
  - SVM models trained using openSMILE



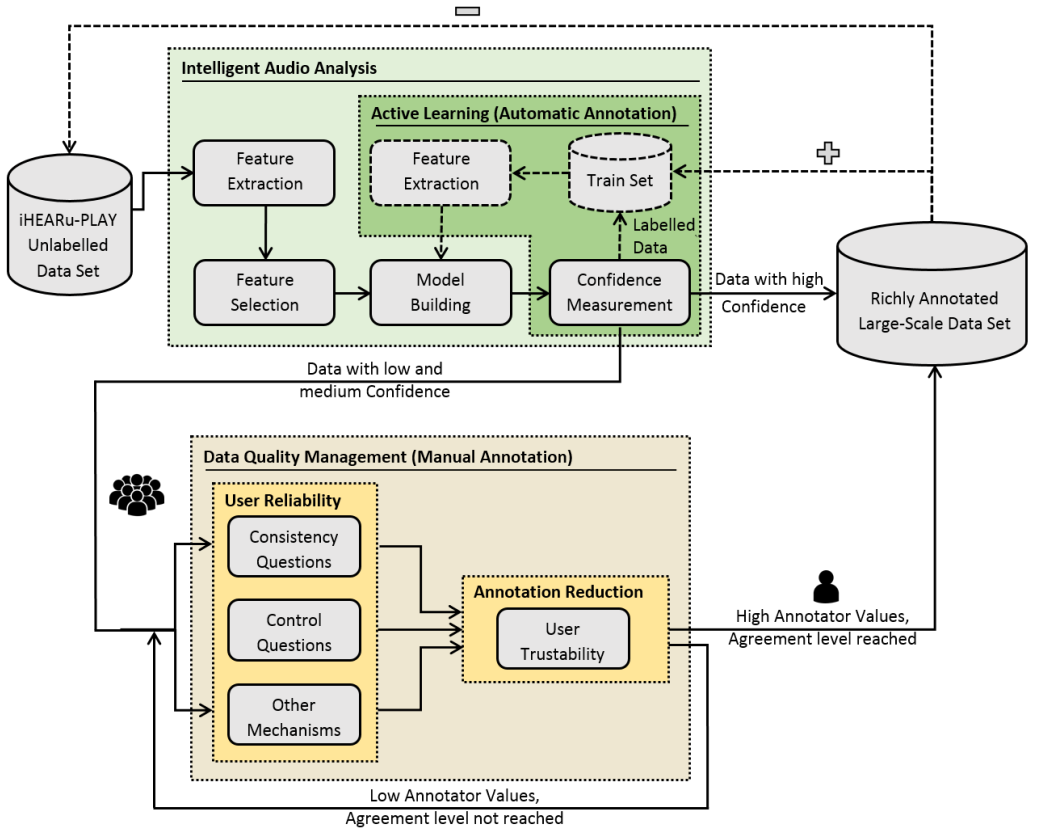
Gamification Aspects

- Player classes, Leader board, badges, bonus items, game score, social aspects,…
- Reduce annotator boredom thus increasing annotation quality



Efficient Annotation

1. Quality control
   - Annotator Trustability
     - Consistency Questions
     - Control Questions
2. Intelligence
   - Novel *active learning* paradigm based on queries to trustworthy annotators

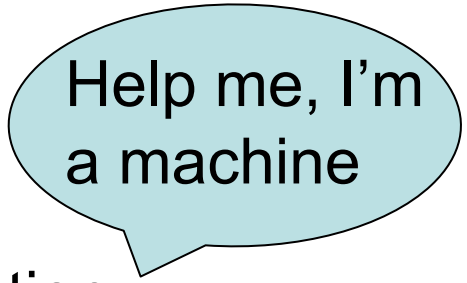


Annotator Trustability

- Combines 3 factors to assign a *Trustability Score*
  1. **Consistency Questions**
     - Questions are repeated to check if the user is paying attention
  2. **Accuracy values**
     - If a user's answer for a particular question is inline with the overall average answer, the annotation deemed trustworthy
  3. **Control Questions**
     - Contain answer possibilities which do not make sense in combination with the given task

Intelligent Labelling

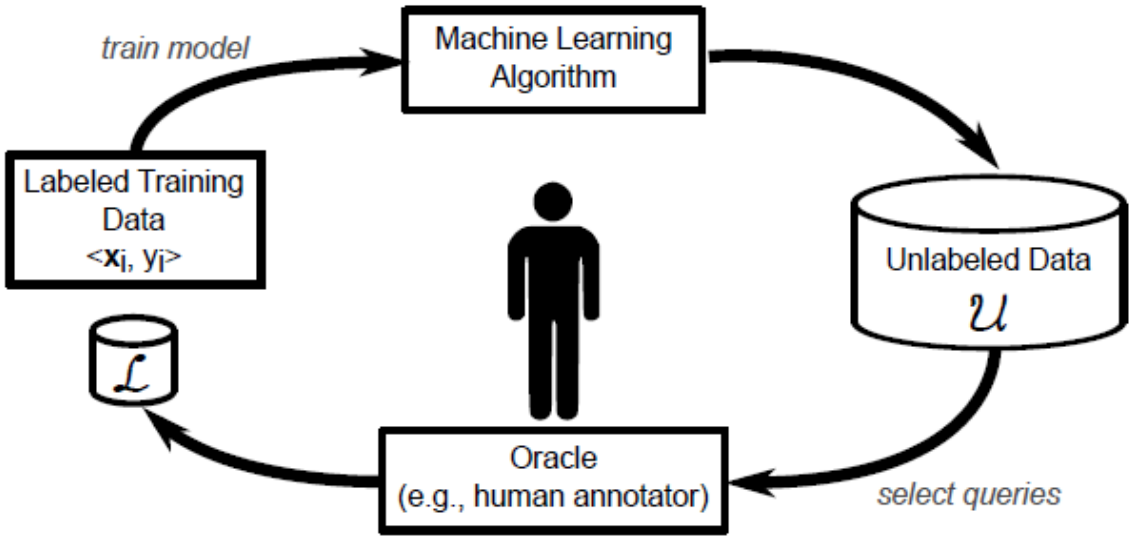
- Use machine learning to reduce the human effort associated to annotation of large corpora



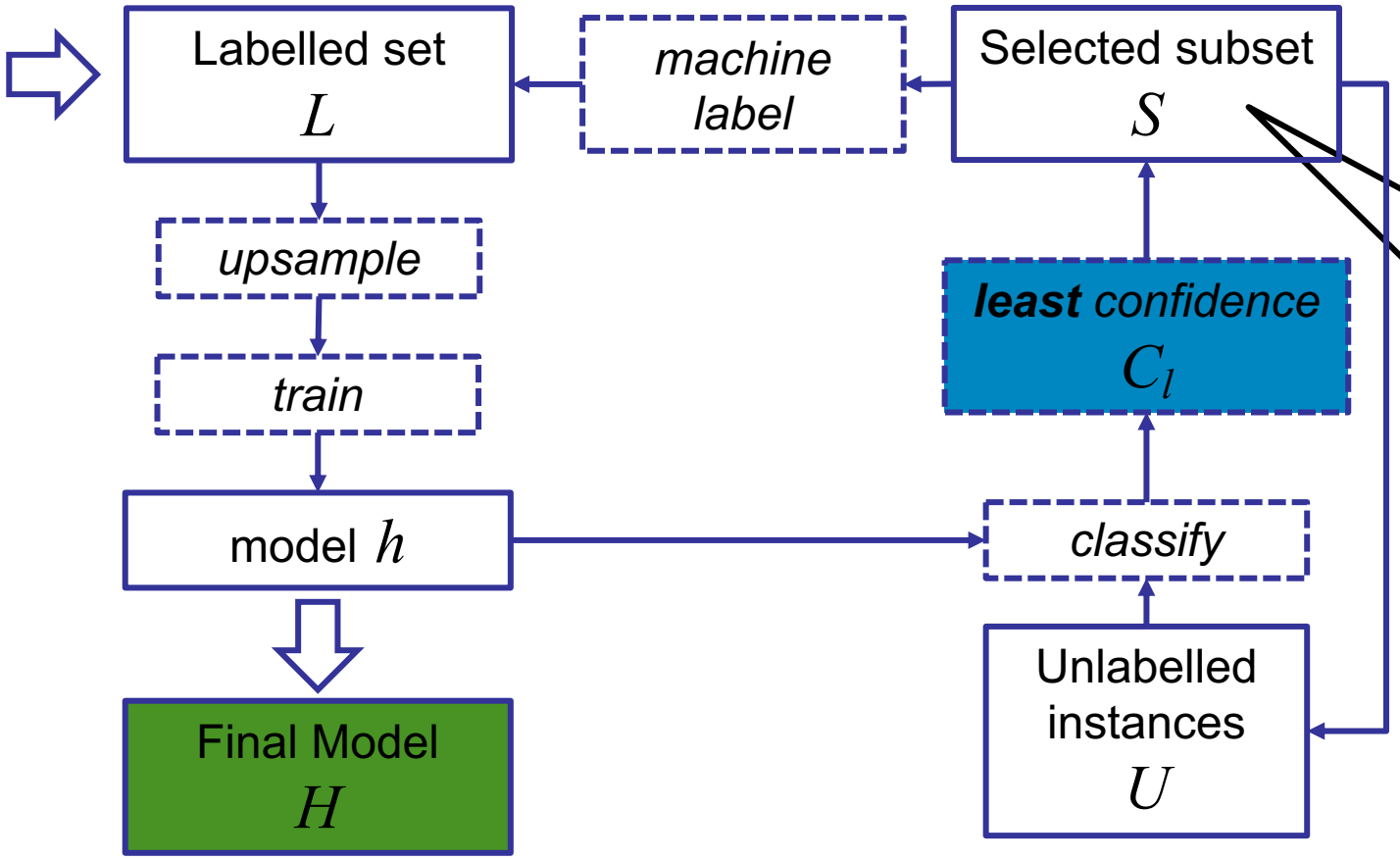
1. Active Learning:
   - Machine finds 'interesting' data for human annotation
2. Semisupervised Learning:
   - After a supervised initialization, a machine labels the data

Active Learning

- The learner chooses the unlabelled data it wishes to have labelled by an oracle (human)



Active Learning



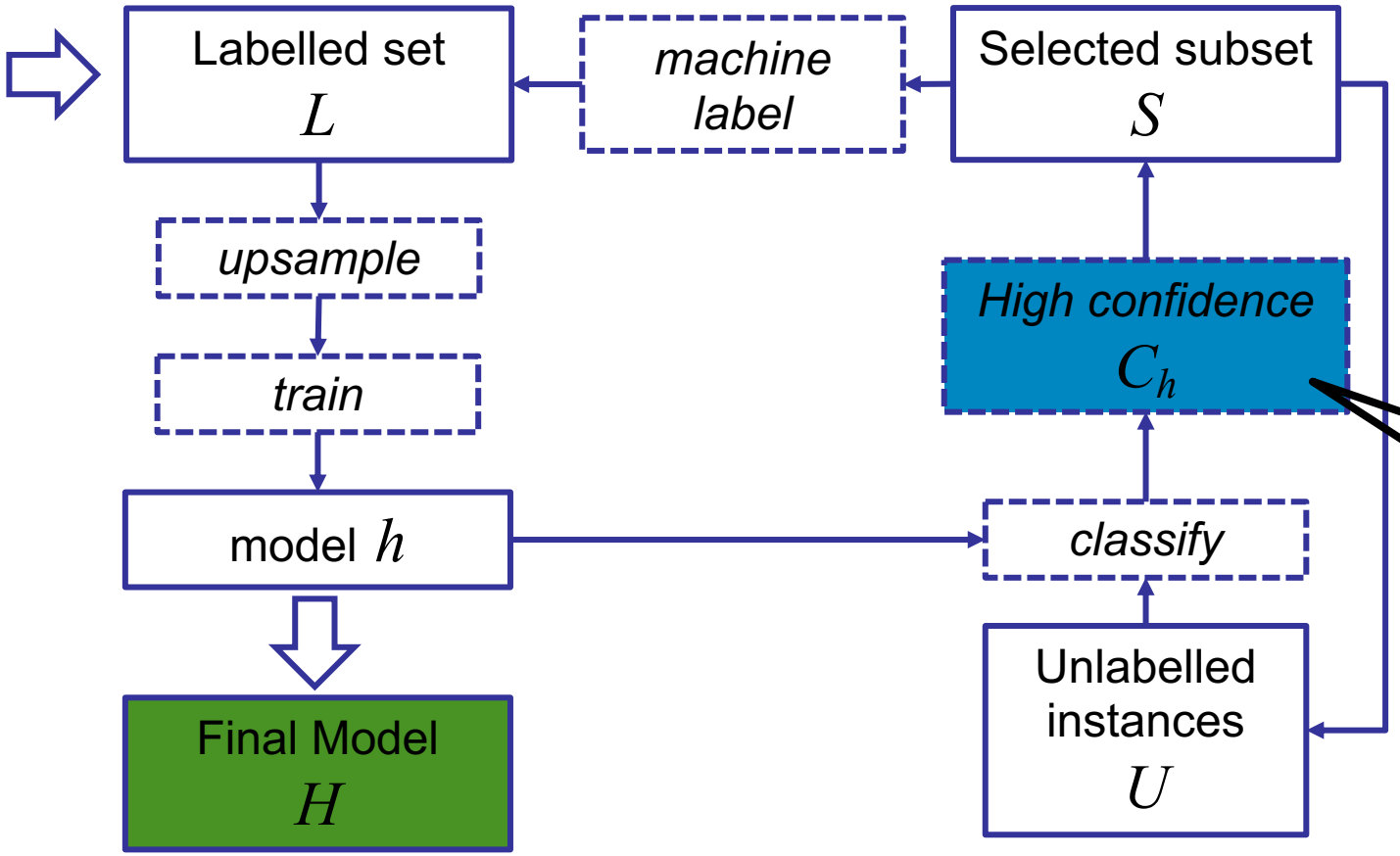
Instances predicted with the **least** confidence values will be selected

$$Q(X) = \begin{cases} 1, if\ X = \dfrac{\arg min}{x} \mid c_x \_ c_l \mid \\ 0, otherwise \end{cases}$$

Semi-Supervised Learning

- Assuming sufficiently robust automatic recognition engine
  - Unlabelled data is classified and integrated into an iterative retraining process.
- Two parameters of interest
  1. Iteration number
     - How often the unlabelled data are relabelled
  2. Upsampling factor
     - Weight the original human-labelled data more strongly than the later added machine-labelled data.

Semi-Supervised Learning



Instances predicted with **high** confidence values will be selected for machine labelling.

$$X = \arg\max_{X \in X} C(X)$$

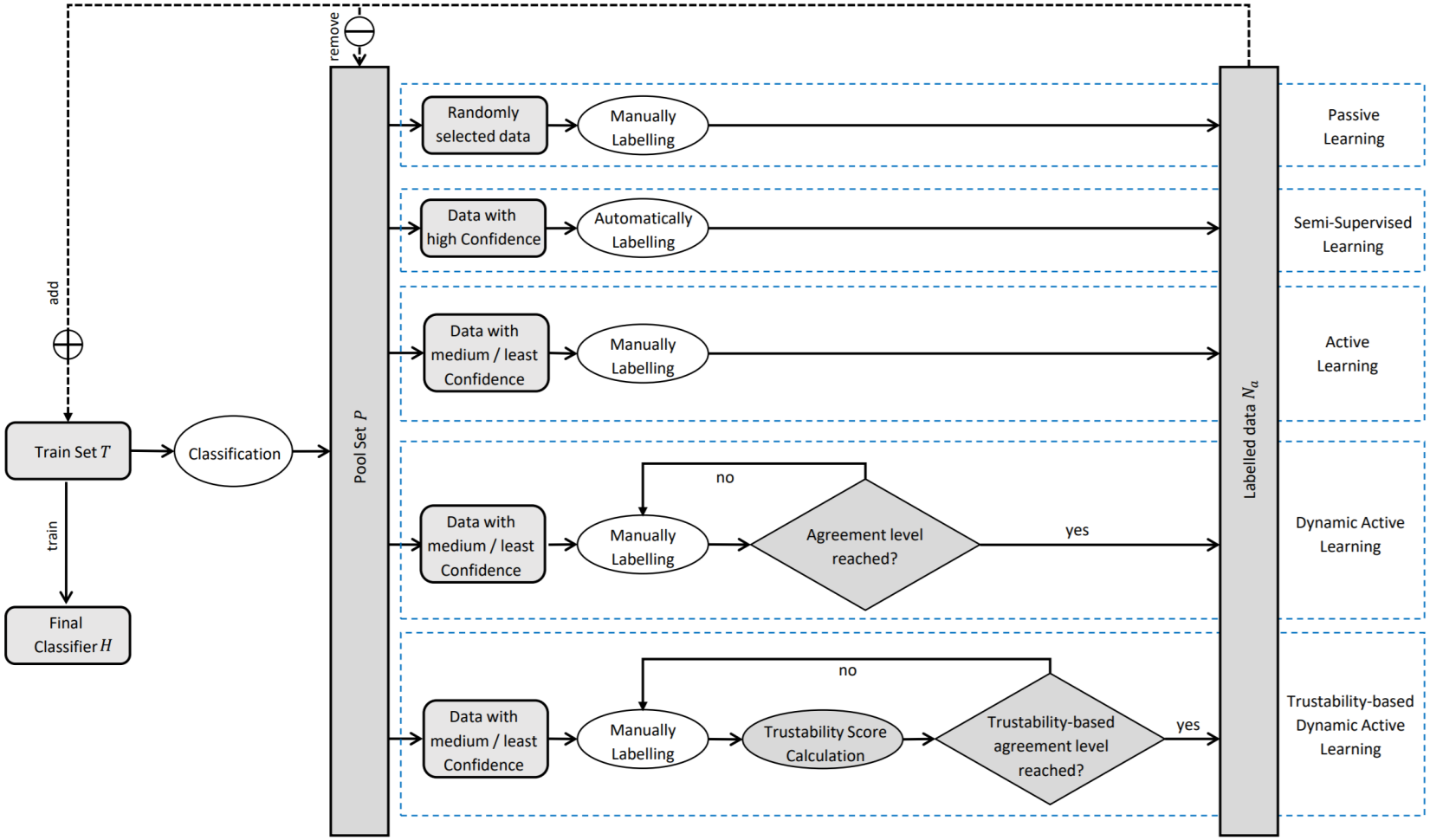
Efficient annotation in iHEARu-PLAY

- Realised using trustability based active learning
  - Dynamic Active learning
    - Conventional approach
      - Stops after a certain number of annotators have been querried
      - Does not take reliability of annotations into account
  - *Trustability based Dynamic Active Learning*
    - Active Learning iterations stop when the user trustability sum of the annotation set reaches a pre-defined limit
    - Explicitly accounts for user trustability

S. Hantke, A. Abstreiter, N. Cummins, and B. Schuller, “Trustability-based Dynamic Active Learning for Crowdsourced Labelling of Emotional Audio Data,” in IEEE Access 13 pages, to appear

Universität Augsburg University

Comparison of annotation approaches:



iHEARu PLAY

iHEARu-PLAY – Intelligent Data Annotation

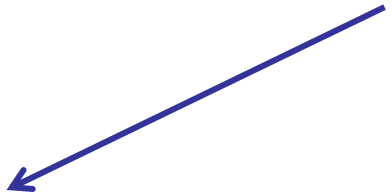
Universität Augsburg University

- Intelligent annotation increase accuracy and save costs
  - FAU-AIBO dataset, 2 emotion classes (Negative, Other)

Algorithm	j	UAR_{max} [%]	*Std.Dev.* [%]	$NA_{max.\ UAR}$	CR [%]	TR [h]
Active Learning						
PL	-	60.03	0.23	48 265	-	-
DAL	2	61.41	0.57	7 453	84.56	3.34
DAL	3	61.50	0.27	11 108	76.99	3.04
TDAL	1	**62.66**	0.86	**4 549**	**90.57**	**3.58**
TDAL	1.5	61.48	0.40	7 012	85.47	3.38
Upsampled Active Learning						
PL	-	72.08	0.28	48 265	-	-
DAL	2	73.20	0.52	19 014	59.77	2.36
DAL	3	72.52	0.36	31 404	33.92	1.38
TDAL	1	**73.71**	0.38	**10 584**	**78.07**	**3.09**
TDAL	1.5	73.04	0.57	19 434	59.73	2.36
Upsampled Active Learning with Semi-Supervised Learning						
PL	-	72.08	0.28	48 265	-	-
DAL	2	72.29	0.39	10 867	77.48	3.06
DAL	3	72.02	0.44	17 027	64.72	2.56
TDAL	1	**72.80**	0.53	**6 051**	**87.46**	**3.46**
TDAL	1.5	72.41	0.46	10 432	78.39	3.14

Proposed TDAL approach

- Achieves the highest UAR with an annotation cost reduction of 87%

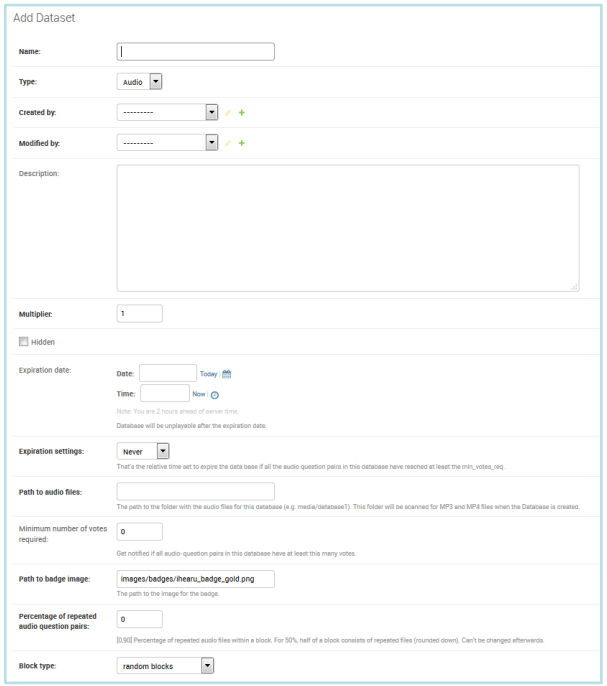


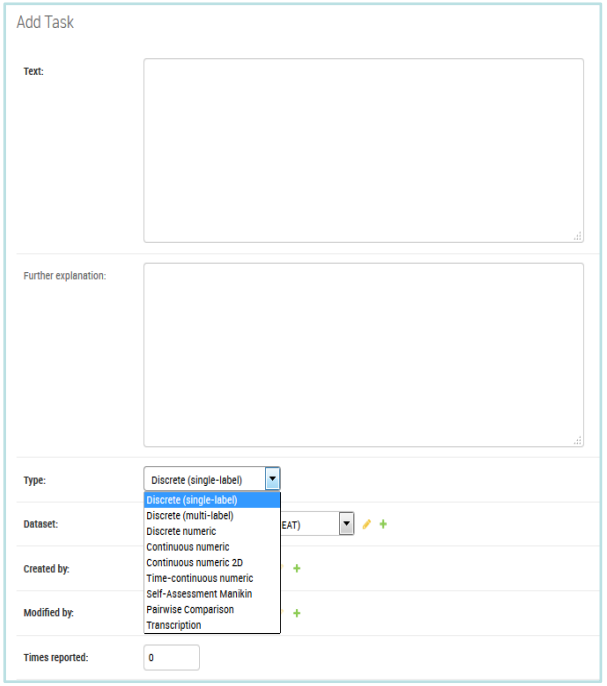
S. Hantke, A. Abstreiter, N. Cummins, and B. Schuller, "Trustability-based Dynamic Active Learning for Crowdsourced Labelling of Emotional Audio Data," in IEEE Access 13 pages, to appear

iHEARu PLAY

Researcher Portal (*Coming very soon*)

- Allows interested researchers to create their own datasets and tasks





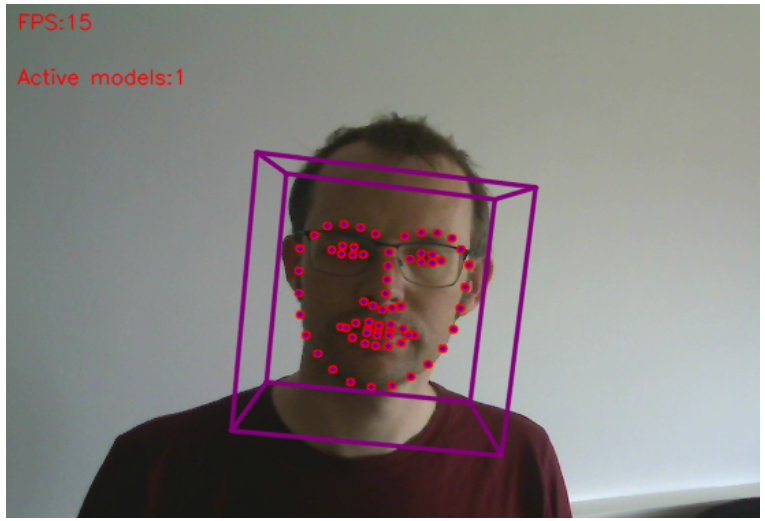
Contents

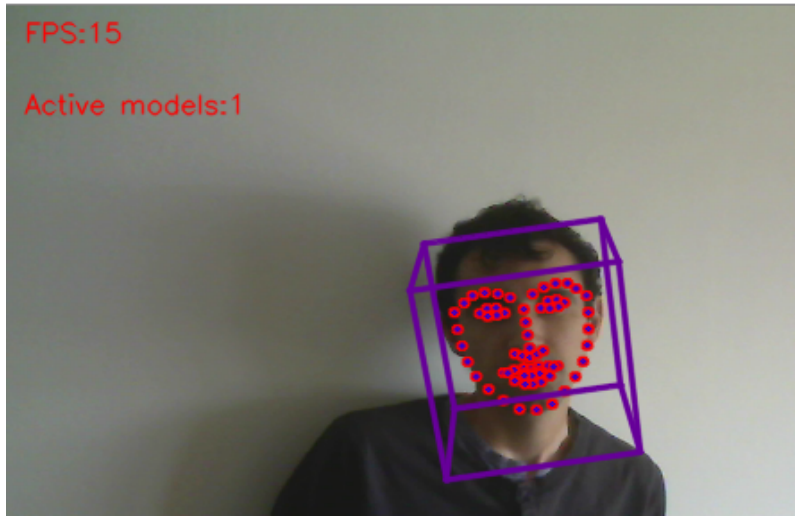
Universität Augsburg University

- CAS²T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- OpenXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

Feature Extractions

- **Abstract representation of raw signal**
  - Extract information relevant to task at hand
  - Reduce redundancies for machine learning





Common Features

Universität Augsburg University

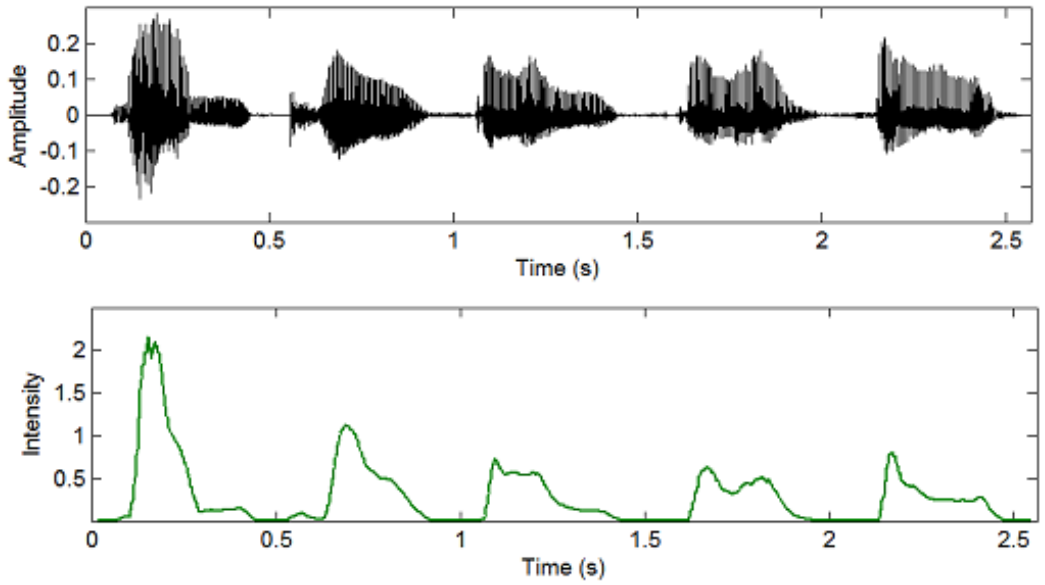
- **The ideal feature**
  - Capture a commonly occurring and easy to measure effect
  - Exhibit large between-class variability, small within class variability
  - Individualised
  - Capture effects specific to health state of interest
  - Predict onset of an affective state
  - Robust to different recording environments
  - Low dimensionality

- **Short-term Energy**
  - Track amplitude envelope of a signal

$$E_m = \sum_n [s(n)w(m-n)]^2$$

$$w(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

  - Window length important

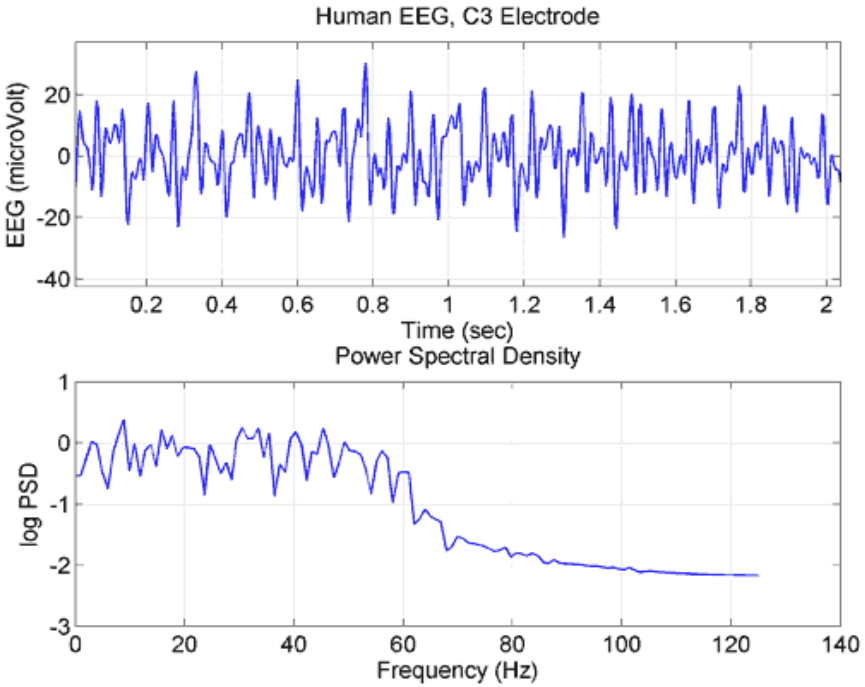


Common Features

- **Power Spectrum**

$$X[k] = \log \left| \sum_{n=0}^{N-1} x[n] e^{-jn\frac{2\pi k}{N}} \right|$$

  - Frequency decomposition
  - Discrete Time Fourier Transform
    - Implemented via FFT

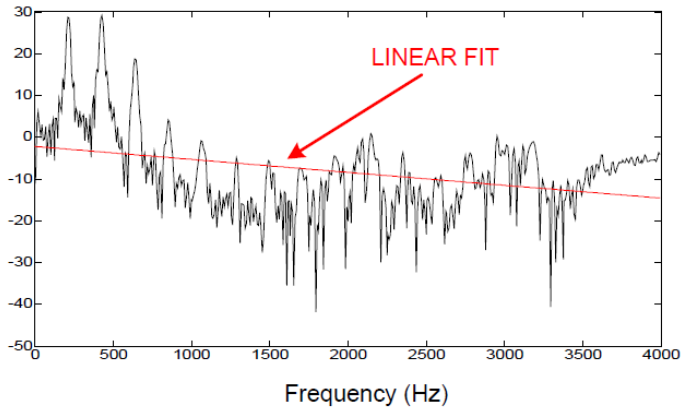


Common Features

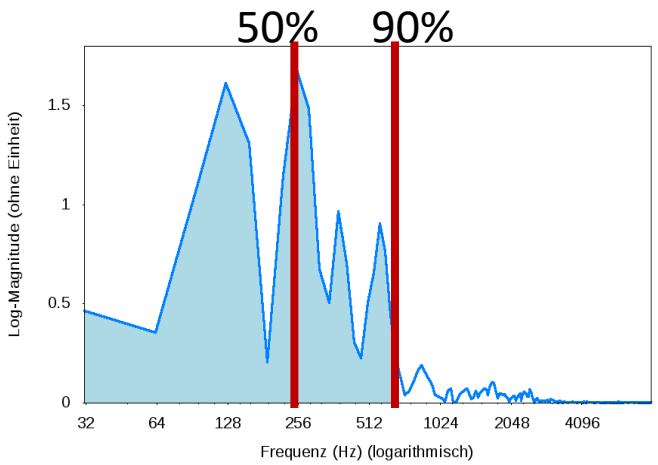
Universität Augsburg University

- **Spectral Gradient** a

$$\hat{y} = am + b$$
$$\min \left\| \underline{y} - \underline{\hat{y}} \right\|_2^2 \rightarrow (a, b)$$



- **Spectral Roll-off-Point**
  - Frequency below which $(X \times 100)\%$ of the total signal energy falls



- **Spectral Entropy**
  - Measure of how noise like signal is

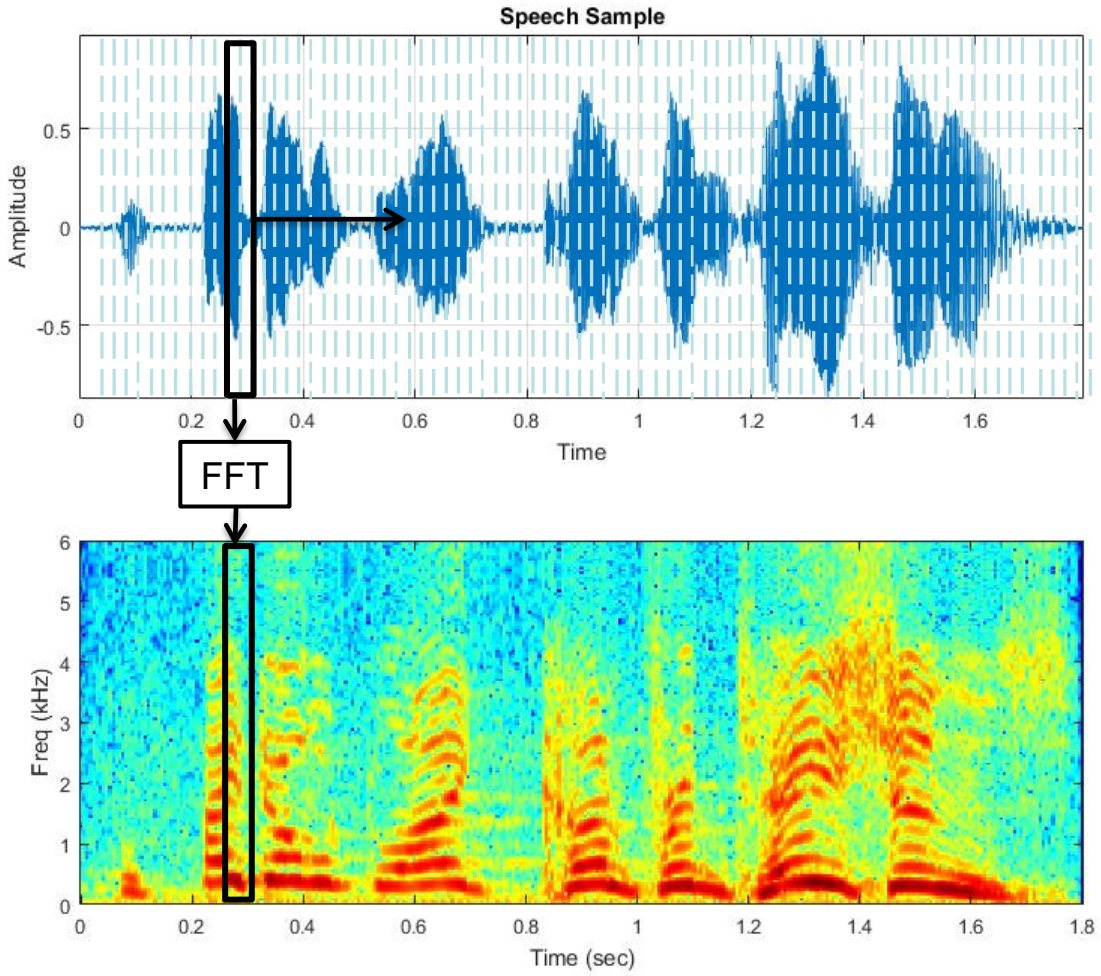
$$S_e = -\sum_{m=m_l}^{m_u} p_x(m) \cdot \log_2 \frac{X(m)}{\sum_{i=m_l}^{m_u} X(i)}$$

Common Features

- **Spectrogram**
  - Short Time Fourier Transform

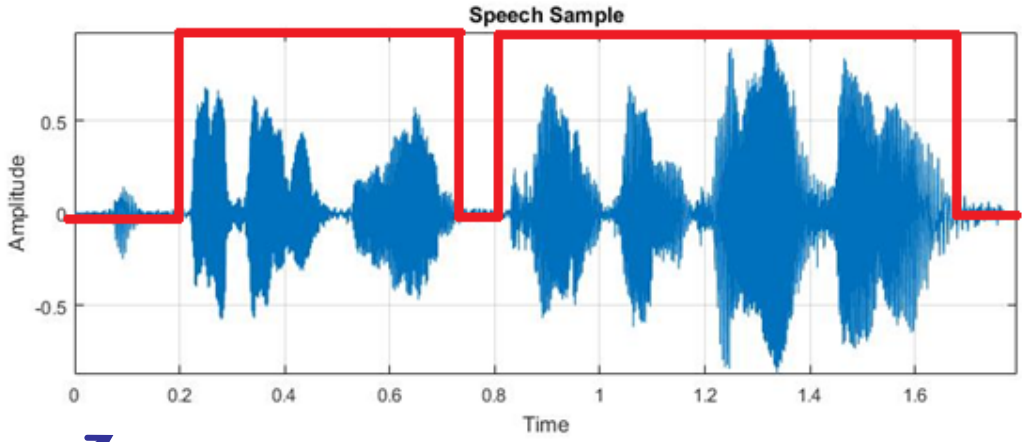
$$S[k,m] = \sum_{n} w[n-m]s[n]e^{-j\frac{2\pi k}{N}n}$$

  - Narrowband (good ΔF, poor ΔT)
  - Wideband (good ΔT, poor ΔF)



Segmentation

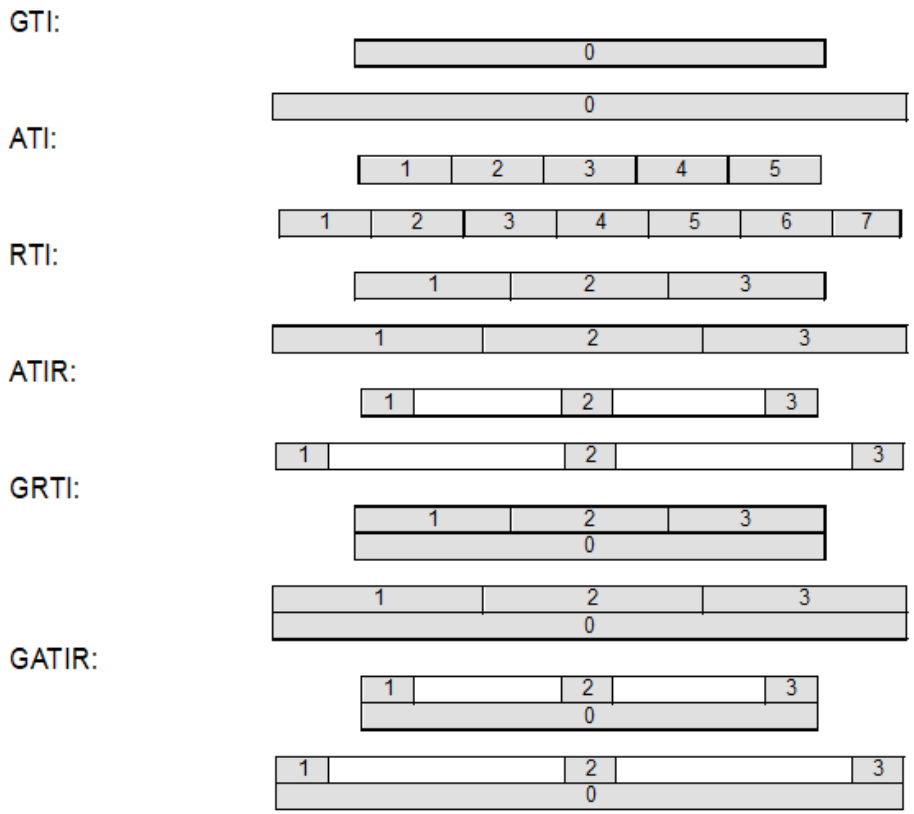
- Phenomena of interest expressed as the evolution of features over time
- Units of Analysis
  - Fixed length
    - Simple to calculate
    - Retains redundant information
  - Dynamic
    - Based on activity detection



Segmentation

Universität Augsburg University

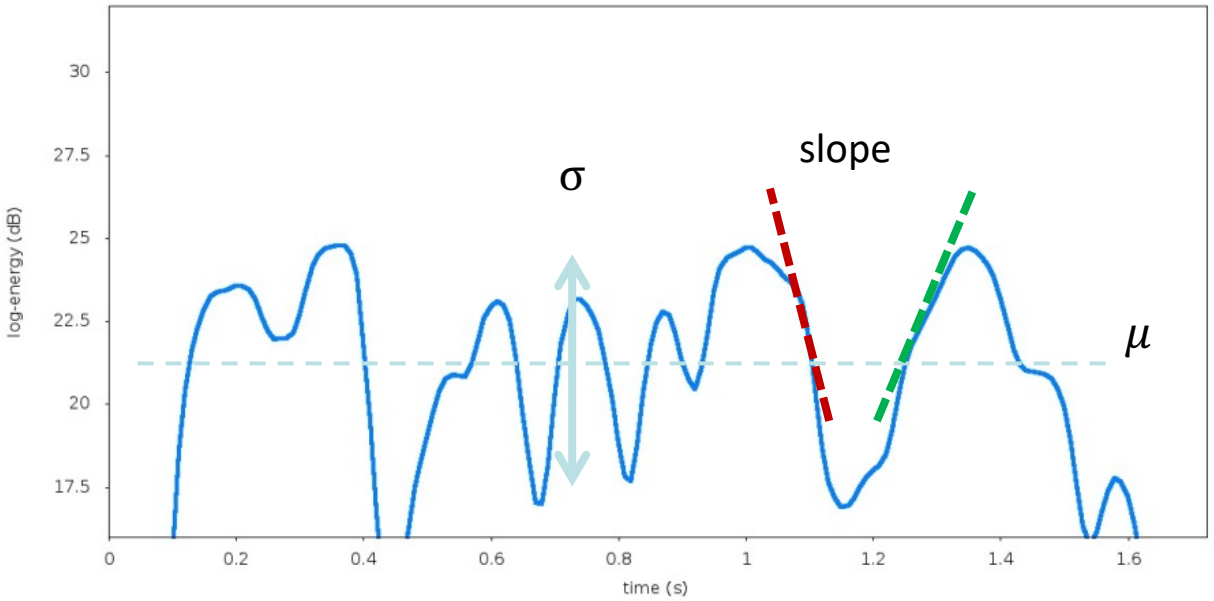
- Fixed Segmentation Schemes
  - Global time interval (GTI)
  - Absolute time intervals (ATI)
  - Relative Time Intervals (RTI)
  - RTI + GTI (GRTI)
  - ATI at relative positions (ATIR)
  - GTI + ATIR (GATIR)
  - ……



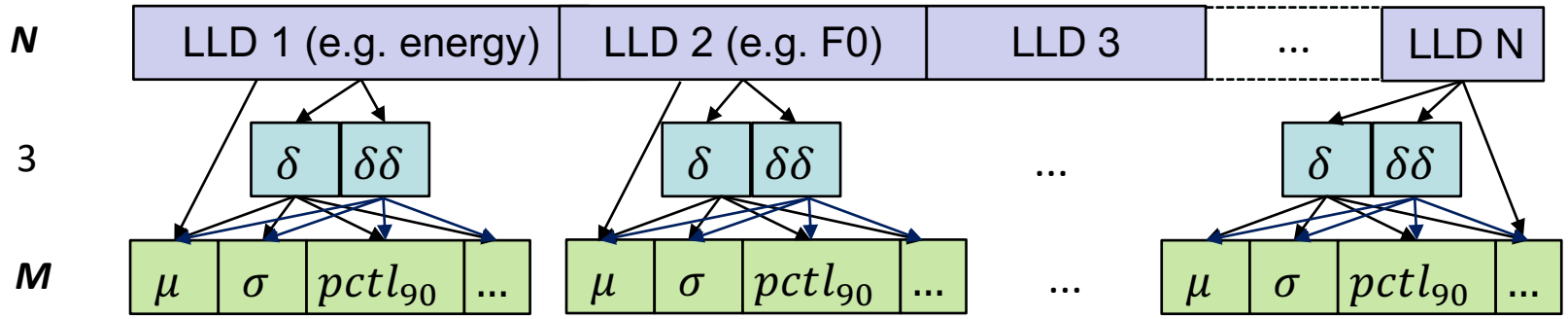
Supra-segmental Features

Universität Augsburg University

- Aim
  - Summarize features over a window in time (e.g., 5 seconds) or the whole signal
  - **Functionals:**
    - Means
    - Moments
    - Extrema
    - Percentile
    - Regression lines
    - Slope
    - …



Generation of large feature spaces



➔ $N \cdot 3 \cdot M$ features!

- Example openSMILE:
  - More than 30 LLD classes implemented
  - over 200 single LLD
  - LLD smoothing and derivations of any order
  - over 80 functionals implemented

Universität Augsburg University

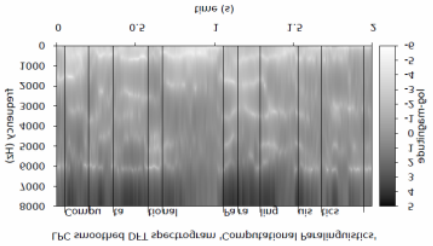
SMILE = **S**peech and **M**ultimedia **I**nterpretation by **L**arge-space **E**xtraction

- Open source
- Unites **audio features** for
  - Speech
  - Music
  - Sound
- **Video features** and supra-segmental features
- Tool **for researchers** and **developers**, not end-users
- Console application
- Free configuration via **configuration files**

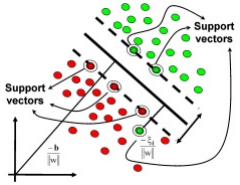
openSMILE

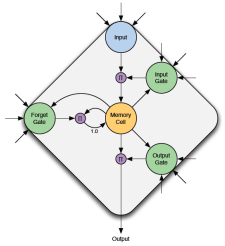
Universität Augsburg University

- **Input/Output:**
  - Files: Audio (wav), Video (openCV), CSV, HTK, ARFF, Text, Libsvm
  - Devices: Soundcard, Webcam
- **Features:**
  - Face and eye location
  - Colour Histograms, Optical Flow, and LBP (from face region)
  - Pitch, Energy, Loudness
  - Jitter, Shimmer, HNR
  - MFCC, PLP, Auditory Spectra, Spectral Statistics (Flux, Centroid, Roll Off, …)
  - LPC, LSP, Formants, Chroma (Pitch Class Profiles), CENS
  - Functionals (summaries): e.g. Means, Moments, Percentiles, Peaks, Temporal
- **Classifiers:**
  - Fast linear SVM (WEKA SMO implementation)
  - (B)LSTM-RNN
  - HMM (Julius LVCSR engine)
  - Voice Activity Detector
- **And many more …**



htk





openSMILE:)
by audEERING™

openSMILE

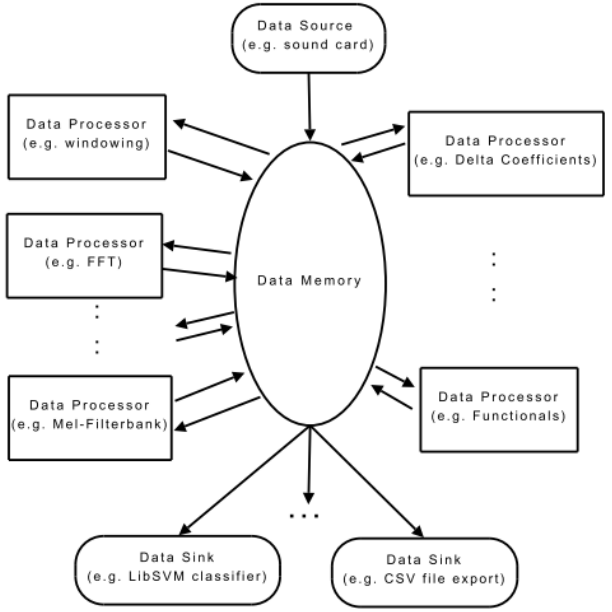
Universität Augsburg University

- Available at:
  - http://audeering.com/technology/opensmile/#download
- Dependencies:
  - Visual Studio 2010 (or higher) - https://www.visualstudio.com
    - If you plan to build your own copy in Windows
  - Android NDK Release (for combining into Android)
  - OpenCV (2.2 or higher) - http://opencv.org
    - If you plan to extract Video features
- Supports:
  - WEKA - http://www.cs.waikato.ac.nz/ml/weka/
  - LibSVM - https://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - HTK - http://htk.eng.cam.ac.uk

openSMILE:) by audEERING™

F. Eyben, F. Weninger, F. Gross, B. Schuller: "Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor", in *Proceedings of ACM Multimedia (MM),* Barcelona, Spain, ACM, ISBN 978-1-4503-2404-5, pp. 835-838, October 2013.

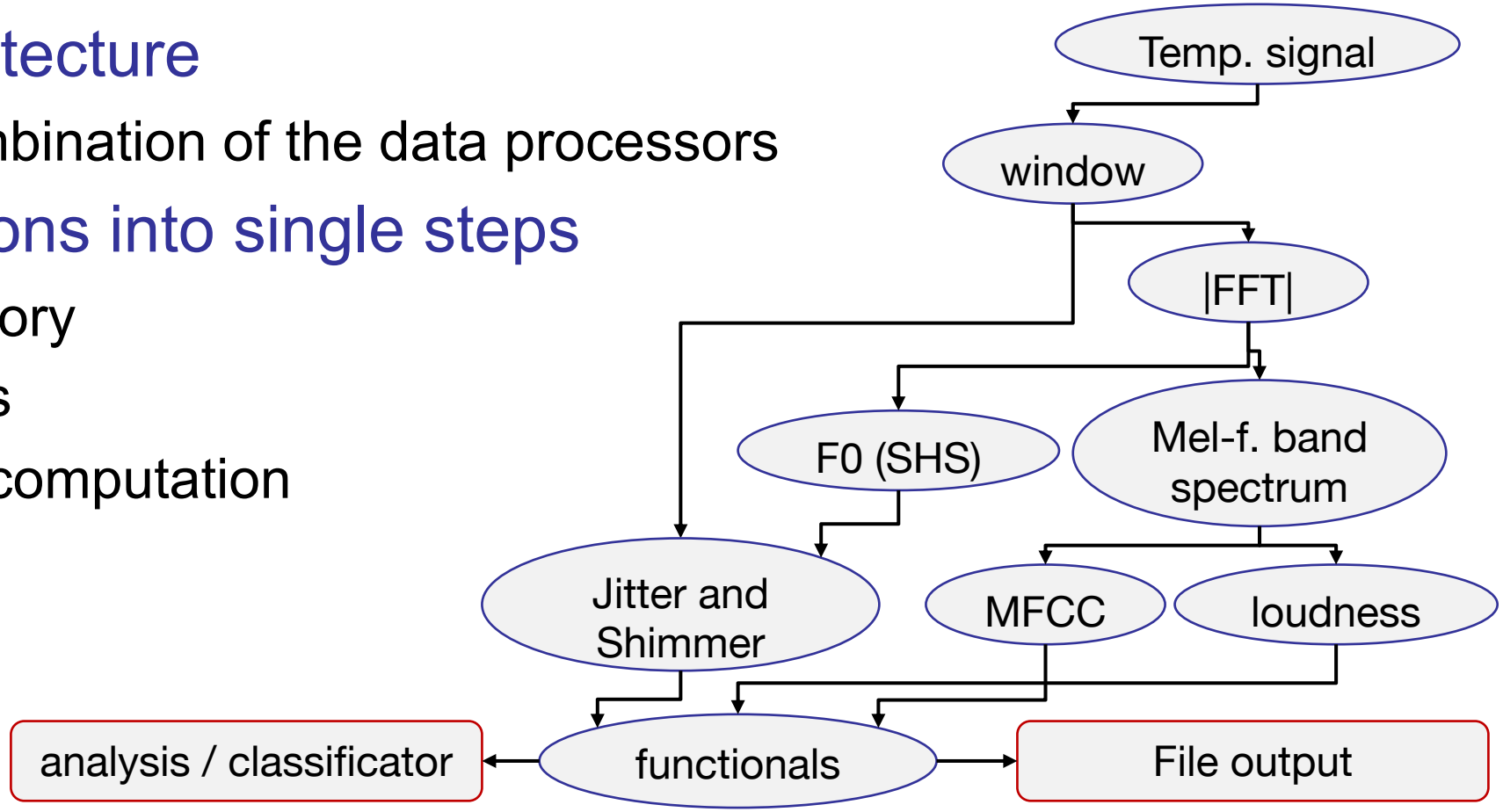
- Data flow handle by a central memory component
  - Cyclic buffers manage data flow between components
  - Manages multiple data memory ‘levels’ internally
  - Data memory location can be written to be one component but read by many
- Basic component types:
  - Data sources
  - Data Processors
  - Data Sinks



openSMILE

Universität Augsburg University

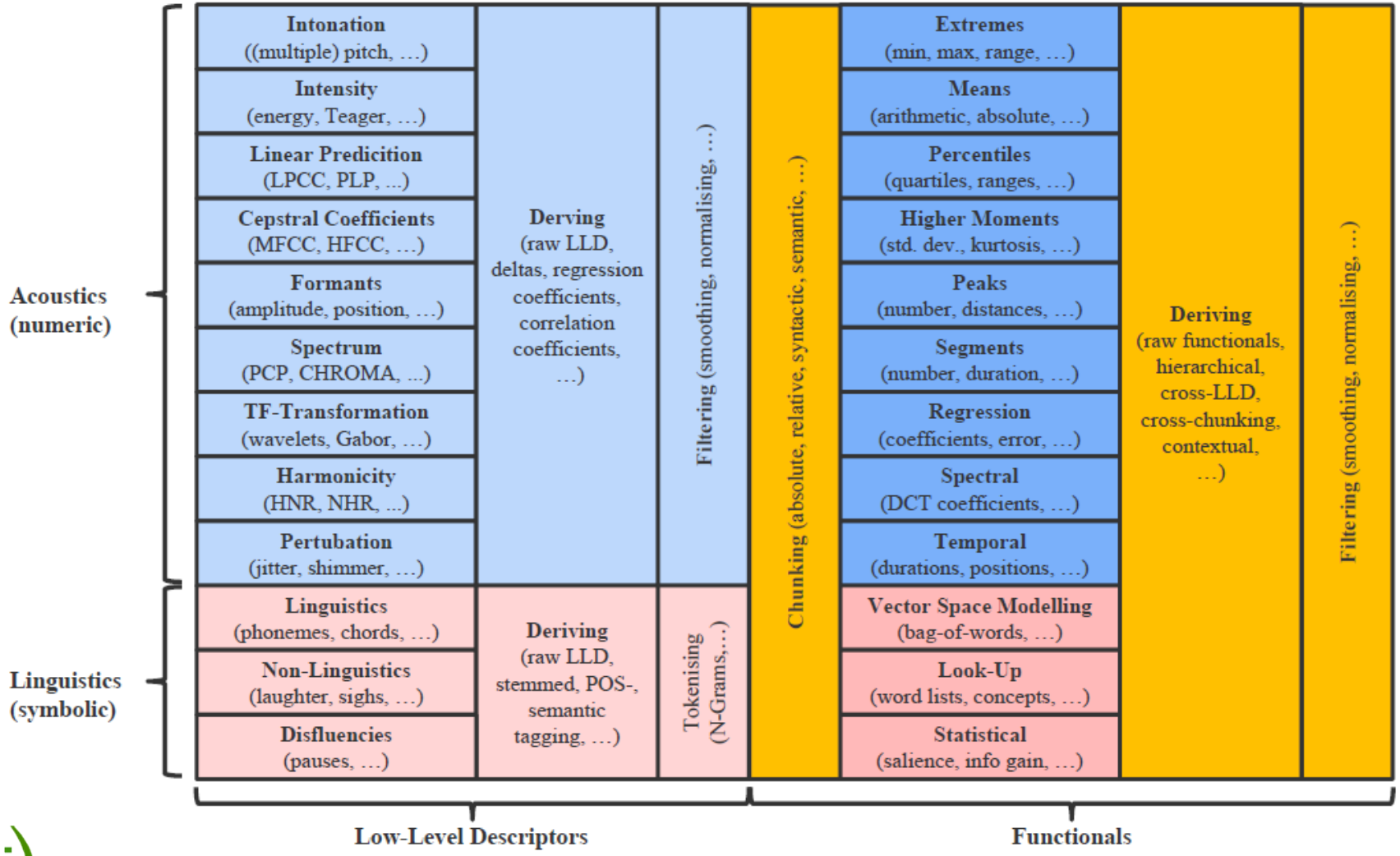
- Modular architecture
  - Arbitrary combination of the data processors
- Split calculations into single steps
  - Shared memory
  - Cyclic buffers
  - Incremental computation



openSMILE:)
by audEERING™

openSMILE

Universität Augsburg University



openSMILE:) by audEERING™

openSMILE

Universität Augsburg University

- Configurations files available for a range of acoustic feature representations
  - extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS)
  - Tailor made for emotion recognition

1 energy related LLD	**Group**
Sum of auditory spectrum (loudness)	Prosodic
25 spectral LLD	**Group**
α ratio (50–1 000 Hz / 1-5 k Hz)	Spectral
Energy slope (0–500 Hz, 0.5–1.5 k Hz)	Spectral
Hammarberg index	Spectral
MFCC 1–4	Cepstral
Spectral Flux	Spectral
6 voicing related LLD	**Group**
F0 (Linear & semi-tone)	Prosodic
Formants 1, 2, (freq., bandwidth, ampl.)	Voice Quality
Harmonic difference H1–H2, H1–A3	Voice Quality
log. HNR, Jitter (local), Shimmer (local)	Voice Quality

openSMILE:) by audEERING™

F. Eyben, K. Scherer, B. Schuller, J. Sundberg, E. Andre,´ C. Busso, L. Devillers, J. Epps, P. Laukka, S. Narayanan, and K. Truong, "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing," IEEE Transactions on Affective Computing, vol. 7, pp. 190–202, April–June 2016.

openSMILE

Universität Augsburg University

- Configurations files available for a range of acoustic feature representations
  - Computational Paralinguistics Challenge feature set (COMPARE)
  - Large brute-force feature set
  - Omnibus feature set for paralingusitics

4 energy related LLD	Group
Sum of auditory spectrum (loudness)	prosodic
Sum of RASTA-filtered auditory spectrum	prosodic
RMS Energy, Zero-Crossing Rate	prosodic
55 spectral LLD	**Group**
RASTA-filt. aud. spect. bds. 1–26 (0-8 k Hz)	spectral
MFCC 1–14 cepstral	cepstral
Spectral energy 250–650 Hz, 1 k–4 k Hz	spectral
Spectral Roll-Off Pt. 0.25, 0.5, 0.75, 0.9	spectral
Spectral Flux, Centroid, Entropy, Slope	spectral
Psychoacoustic Sharpness, Harmonicity	spectral
Spectral Variance, Skewness, Kurtosis	spectral
6 voicing related LLD	**Group**
F0 (SHS & Viterbi smoothing)	prosodic
Prob. of voicing	voice quality
log. HNR, Jitter (local & DDP), Shimmer (local)	voice quality

B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism," in Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, (Lyon, France), pp. 148–152, ISCA, ISCA, August 2013.

openSMILE:) by audEERING™

- Running from command line:

```
>opensmile-2.3.0\bin\Win32\SMILExtract_Release
-C config/gemaps/GeMAPSv01a.conf
-I …\wav\input.wav
-O …\outputfiles\output.arff
-class label
```

Path to OpenSMILE Executable

Path to OpenSMILE Config file

Paths to input and output file

Input file label (if know)

- Configuration files
  - Construct your own feature sets
  - INI-style le format:

```
[instancename:configType] <-- this specifies the header
variable1 = value <-- example of a string variable
variable2 = 7.8 <-- example of a "numeric" variable
variable3 = X <-- example of a "char" variable
subconf.var1 = myname <-- example of a variable in a sub type
myarr[0] = value0 <-- example of an array
myarr[1] = value1
anotherarr = value0;value1 <-- example of an implicit array
noarray = value0\;value1 <-- use \; to quote the separator ';'
strArr[name1] = value1 <-- associative arrays, name=value pairs
strArr[name2] = value2
; line-comments may be expressed by ; // or # at the beginning
```

- Configuration files
  - Enabling components:

```
[componentInstances:cComponentManager]
instance[dataMemory].type=cDataMemory
instance[waveIn].type=cWaveSource
instance[frame60].type=cFramer
instance[win60].type=cWindower
instance[fft60].type=cTransformFFT
instance[fftmp60].type=cFFTmagphase
instance[lld].type=cCsvSink
printLevelStats=0
```

always required

always required

always required

- Configuration files

```
[waveIn:cWaveSource]
writer.dmLevel=wave
filename=\cm[inputfile(I){test.wav}:input
file]
monoMixdown=1

[frame60:cFramer]
reader.dmLevel = wave
writer.dmLevel = frame60
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

[win60:cWindower]
reader.dmLevel = frame60
writer.dmLevel = winG60
winFunc = gauss
gain = 1.0
sigma = 0.4
```

```
[fft60:cTransformFFT]
reader.dmLevel = winG60
writer.dmLevel = fftcG60

[fftmp60:cFFTmagphase]
reader.dmLevel = fftcG60
writer.dmLevel = fftmagG60
phase = 0

[lld:cCsvSink]
reader.dmLevel = fftmagG60
filename=\cm[lld{O}:output csv file]
append = 0
timestamp = 1
printHeader = 1
```

→ **openSMILE book**

(*see tutorial material*)

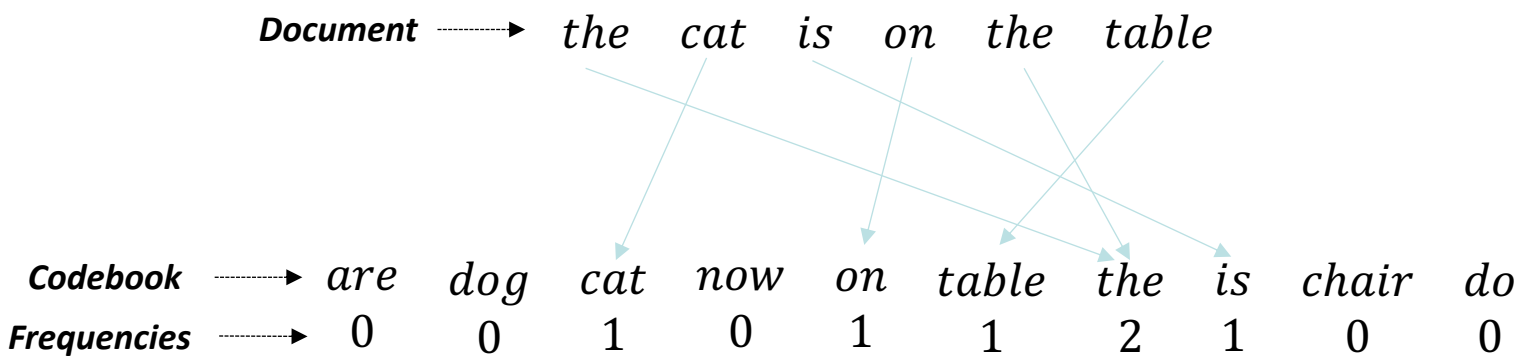
Contents

Universität Augsburg University

- CAS²T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- openXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

Bag-of-Words

- Originally developed for natural language processing
  - Documents are represented as an unordered word frequency taken from a codebook



Bag-of-X-Words

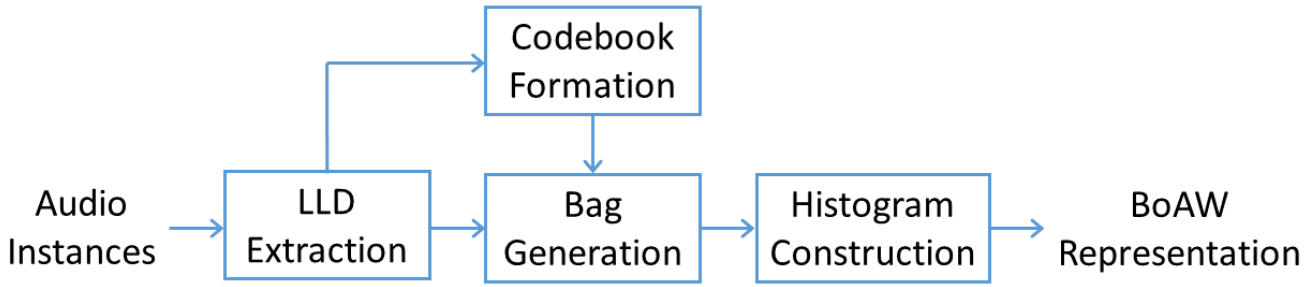
Universität Augsburg University

- BoW has been adopted by the audio and visual communities
  - Bag-of Audio Words (BoAW)
    - Frame level acoustic LLDs (i.e., MFCCs) are extracted from the audio signal and the quantised according to a codebook (dictionary)
  - Bag-of Visual Words (BoVW)
    - Local image features extracted from an image and quantised according to a codebook

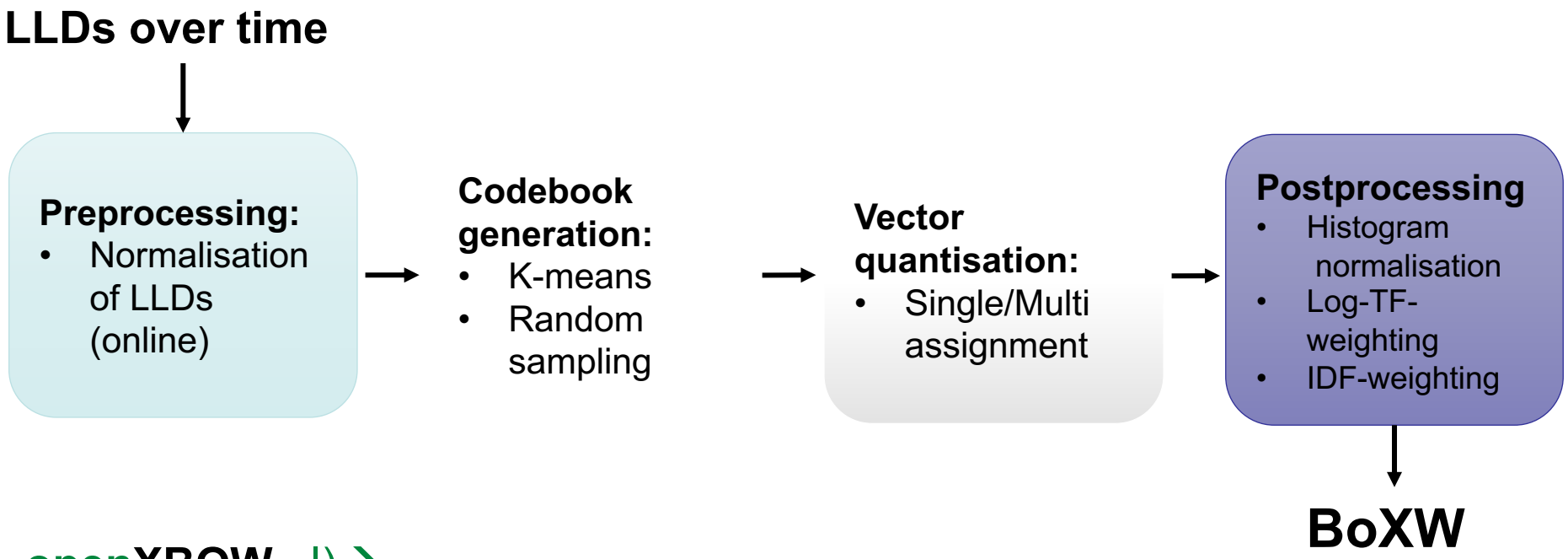
Bag-of-X-Words

Universität Augsburg University

- Each frame-level LLD vector is assigned to an audio word from a previously learnt codebook
  - Counting the number of assignments for each audio word, a fixed length histogram (bag) is generated
- Key Parameters
  - Size of codebook (C_s)
  - Number of Assignments (N_a)



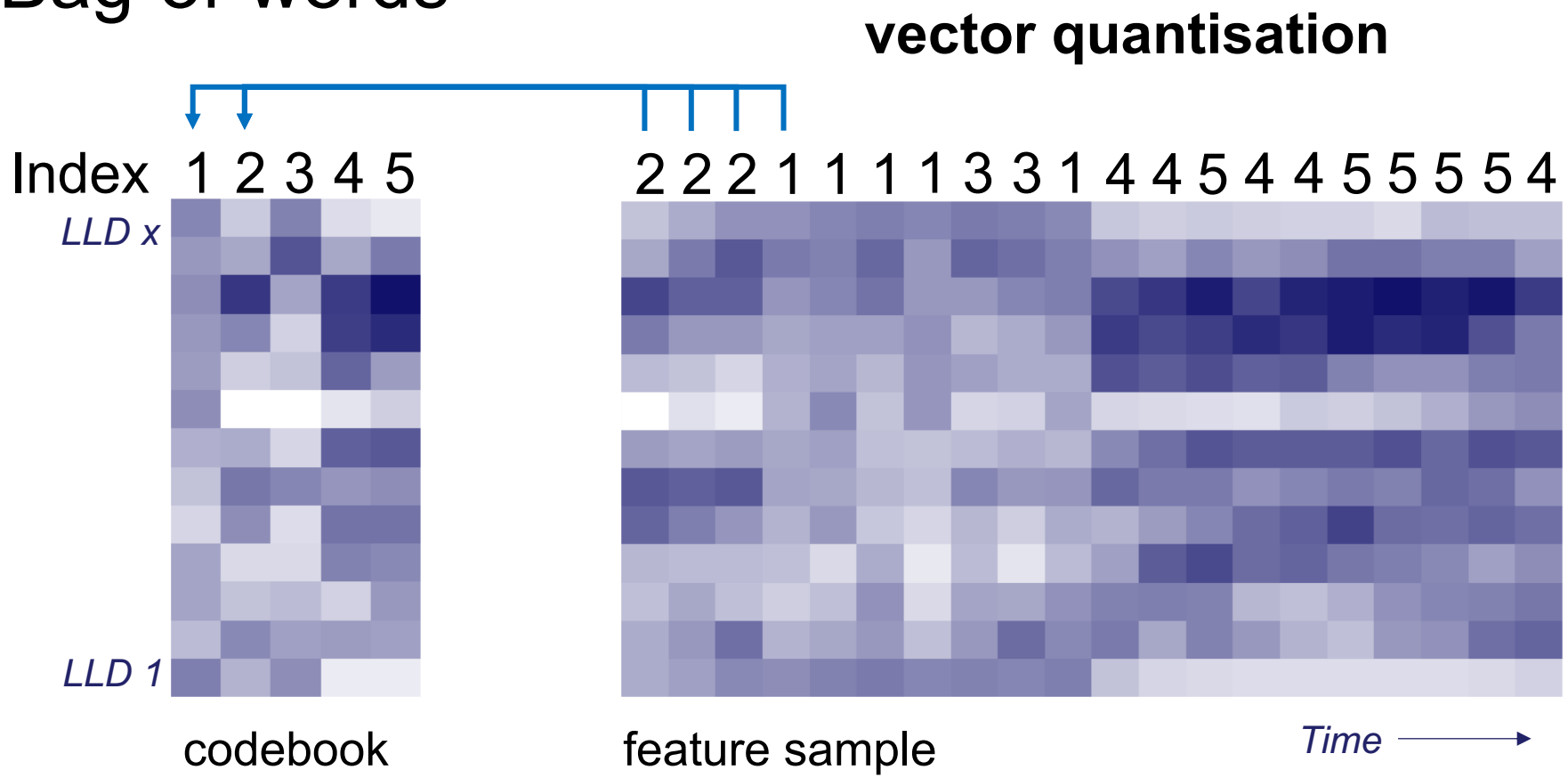
Vector Quantisation



openXBOW –|)→

Vector Quantisation

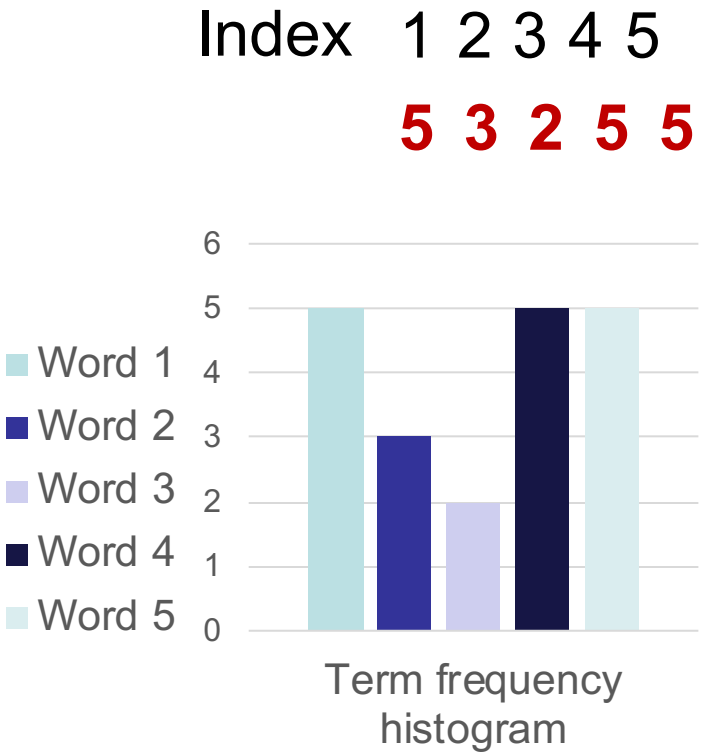
- Bag-of-words

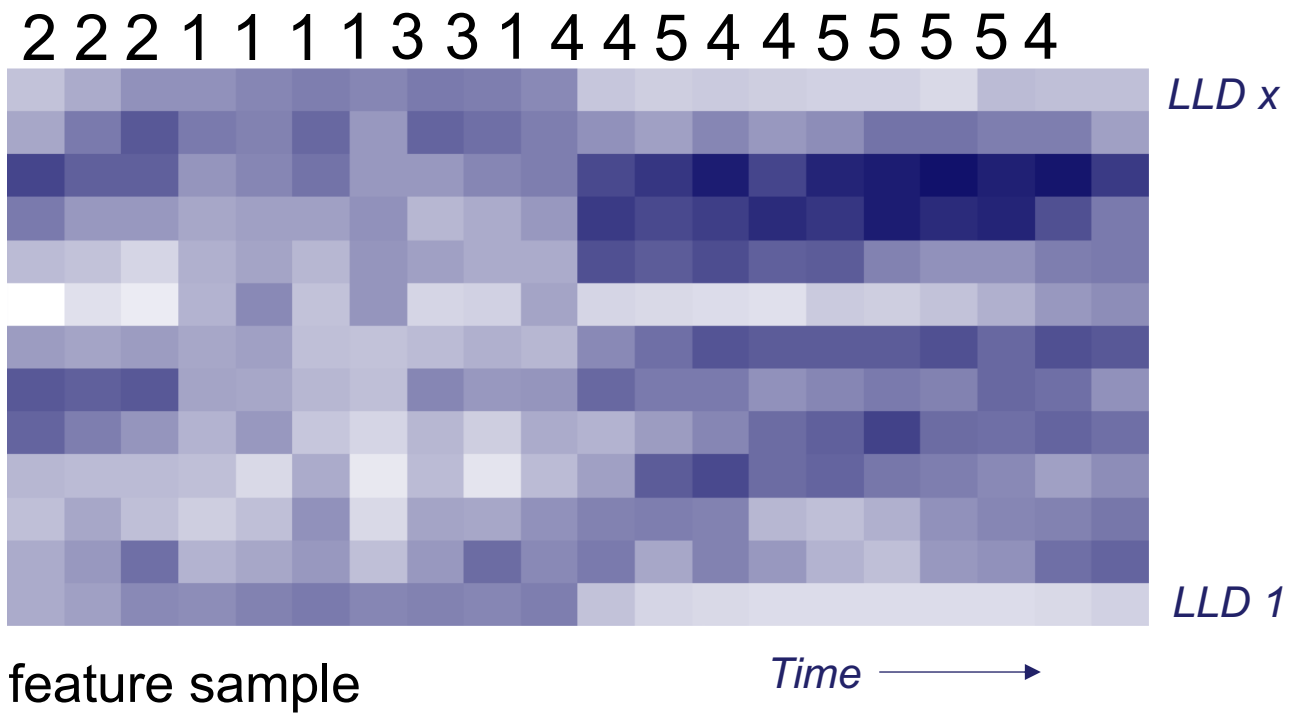


Vector Quantisation

Universität Augsburg University

- Bag-of-Words

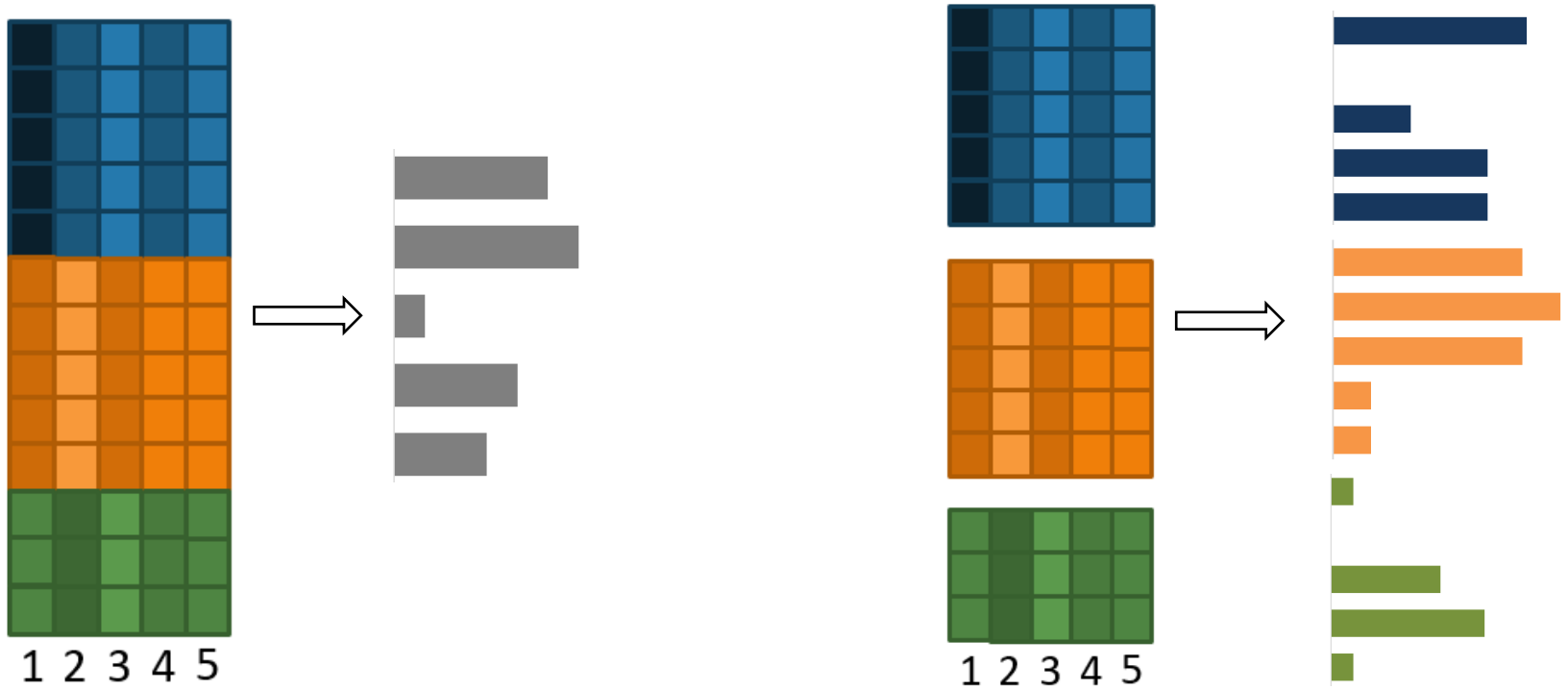




Vector Quantisation

- Bag-of-words

So far: ONE codebook for all LLDs → One per feature type



Bag-of-X-Words

- Advantages
  - Robustness
    - Quantisation step allows this technique to have a degree of tolerance to small perturbations in the input data
  - Privacy
    - Abstract/Sparse representation of the original feature space
  - Time invariance
    - Generates a fixed length vector regardless of time
    - Normalise according to the time/length of the file
  - Multimodal Fusion
    - Different modalities are quantised according to their own codebook
    - This overcomes issues relating to different sampling rates

Bag-of-X-Words

Universität Augsburg University

- State-of-the-Art Results of emotion prediction
  - RECOLA dataset

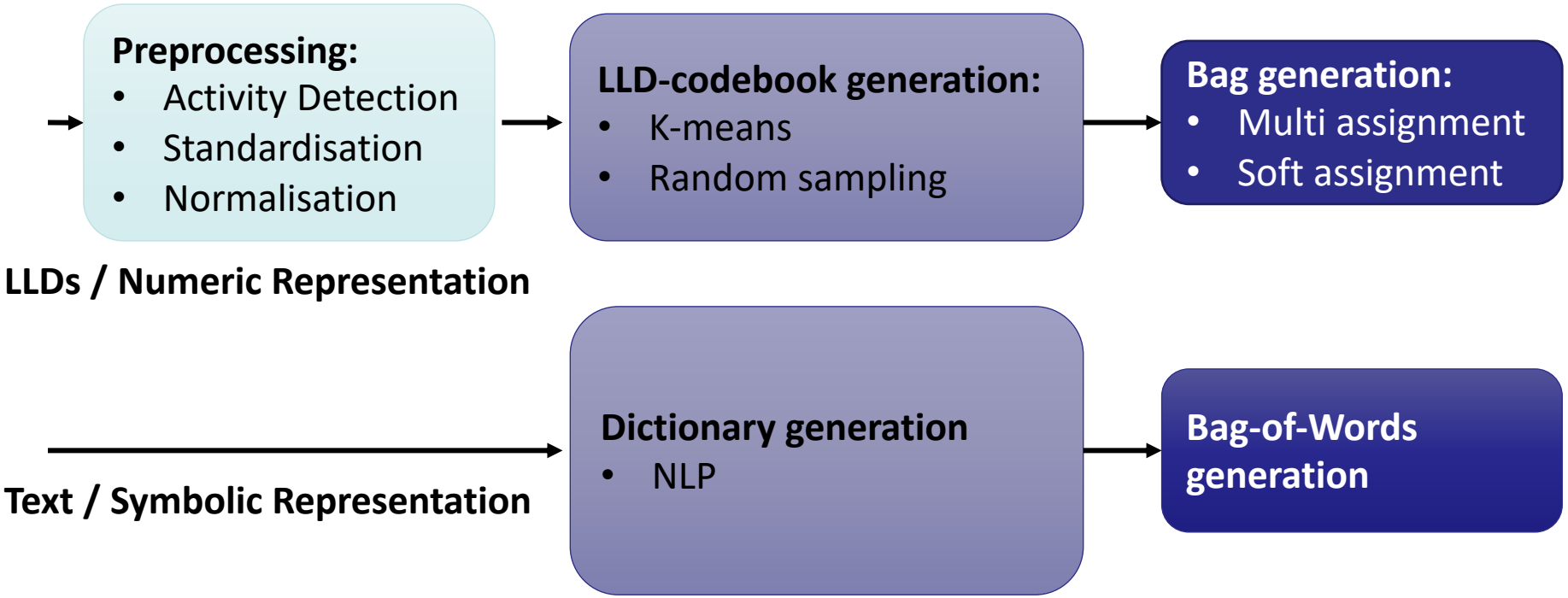
Model	Reference	Arousal		Valence	
		CCC Valid	CCC Test	CCC Valid	CCC Test
BLSTM	He et al. (2015)	.800		.398	
End-to-end (CNN + BLSTM)	Trigeorgis et al. (2015)	.741	.686	.325	.261
Functionals	Baseline	.790	.720	.459	.402
BoAW	Proposed method	.793	.753	.550	.430
Functionals + BoAW	Proposed method	.799	.738	.521	.465

M. Schmitt, F. Ringeval, and B. Schuller, “At the Border of Acoustics and Linguistics: Bag-of-Audio-Words for the Recognition of Emotions in Speech,” in *Proceedings INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*, (San Francisco, CA), pp. 495–499, ISCA, September 2016.

openXBOW

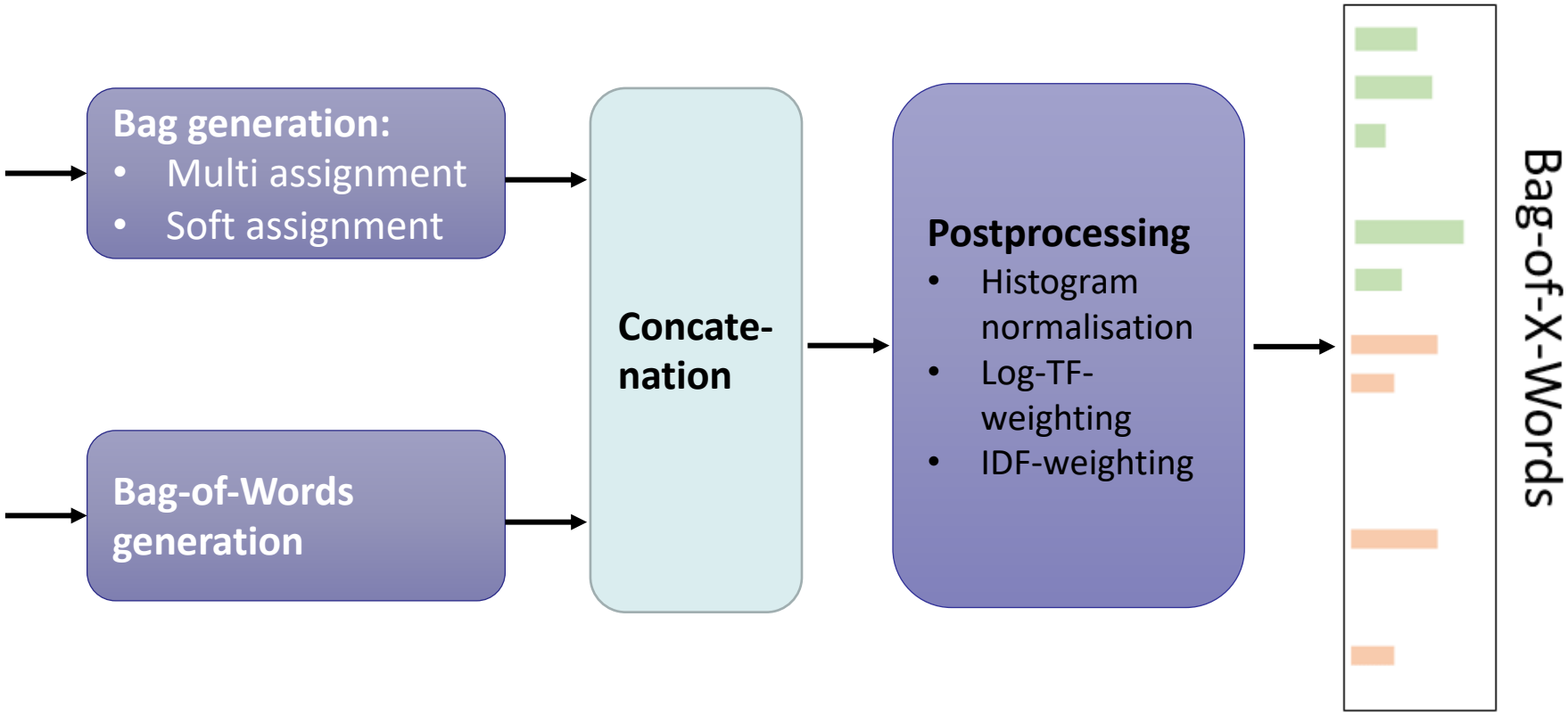
Universität Augsburg University

- Open-source toolkit for the generation of BoW representations across modalities
- Implemented in Java
  - Can be used on any platform
- Supports ARFF (weka), CSV, and LibSVM file formants


Preprocessing:
• Activity Detection
• Standardisation
• Normalisation
LLD-codebook generation:
• K-means
• Random sampling
Bag generation:
• Multi assignment
• Soft assignment
LLDs / Numeric Representation
Dictionary generation
• NLP
Bag-of-Words generation
Text / Symbolic Representation

openXBOW

Universität Augsburg University



openXBOW -|)->

iHEARu-EAT

Universität Augsburg University

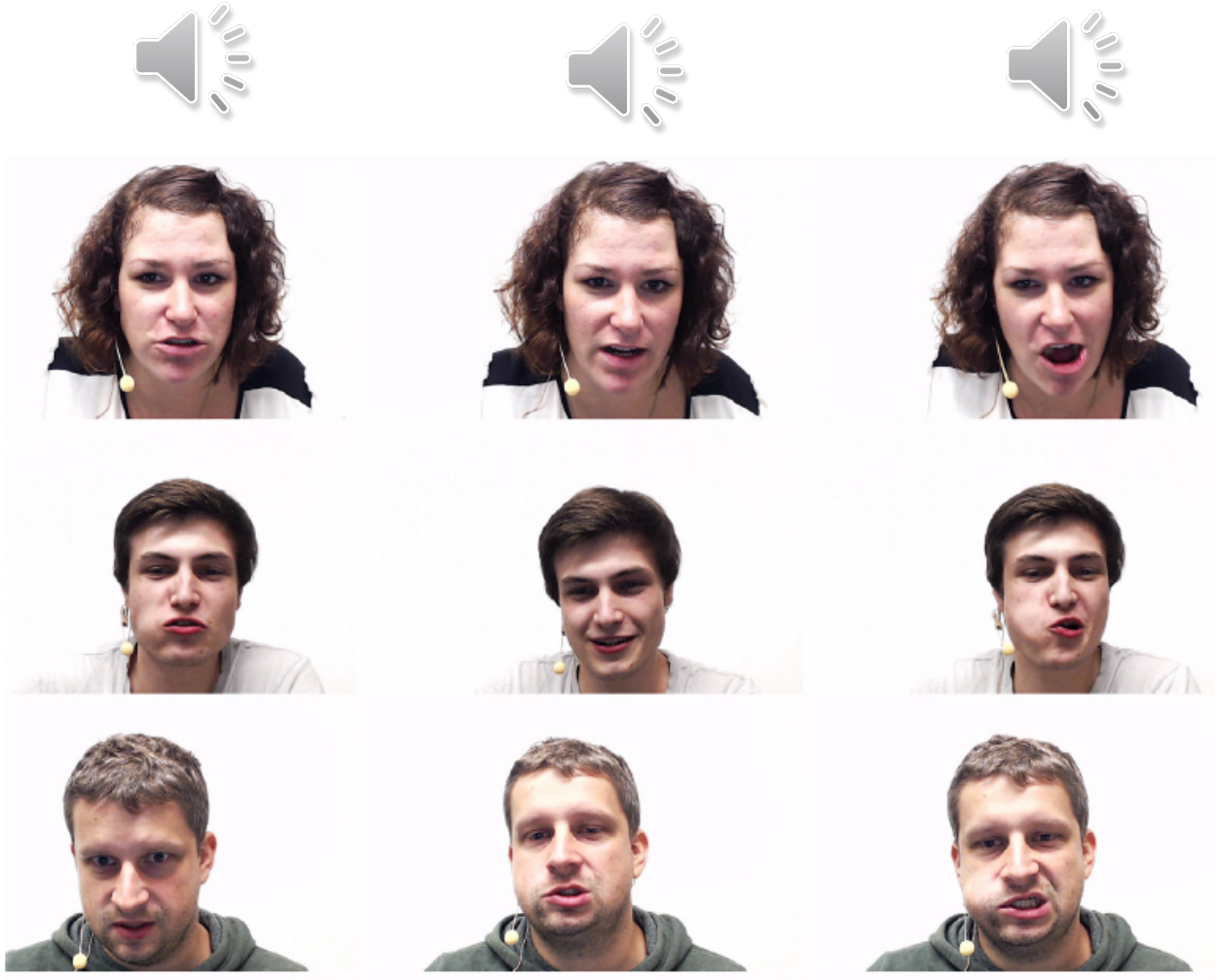
- OpenXBOW tutorial based on iHEARu-EAT data
  - Release as part of ComParE 2015
  - 30 subjects (15 f, 15 m; 26.1±2.7 years)
  - 27 German; 1 Chinese, 1 Indian, 1 Tunisian origin

#	Train	Test	Σ
No Food	140	70	210
Apple	140	56	196
Nectarine	133	63	196
Banana	140	70	210
Crisp	140	70	210
Biscuit	133	70	203
Gummi bear	119	70	189
Σ	945	469	1 414

S. Hantke, F. Weninger, R. Kurle, F. Ringeval, A. Batliner, A. El-Desoky Mousa, and B. Schuller, "I Hear You Eat and Speak: Automatic Recognition of Eating Condition and Food Types, Use-Cases, and Impact on ASR Performance," *PLoS ONE*, vol. 11, pp. 1–24, May 2016

iHEARu-EAT

Universität Augsburg University



No-Food Banana Crisp

openXBOW

Universität Augsburg University

- Available at:
  - https://github.com/openXBOW/openXBOW/
- Dependencies:
  - JAVA
    - https://java.com/en/download/
- Supports:
  - WEKA
    - http://www.cs.waikato.ac.nz/ml/weka/
  - LibSVM
    - https://www.csie.ntu.edu.tw/~cjlin/libsvm/

M. Schmitt and B. Schuller, “openXBOW – Introducing the Passau Open-Source Crossmodal Bag-of-Words Toolkit,” *Journal of Machine Learning Research, vol. 18*, 2017. 5 pages, to appear

- All options with a corresponding help text are displayed with command line argument

```
java -jar openXBOW.jar -h
```

  - All configurations are made via command line
  - Note, please install JAVA

- **Creating a BoAW representations**
  - LLDs in .arff or .csv formant
    - 1st column must contain file name
    - 2nd column must contain frame time stamp
    - Last column can contain the target class

```
@relation openSMILE_features

@attribute name string
@attribute frameTime numeric
@attribute pcm_fftMag_mfcc[1] numeric
@attribute pcm_fftMag_mfcc[2] numeric
@attribute pcm_fftMag_mfcc[3] numeric
@attribute pcm_fftMag_mfcc[4] numeric
@attribute pcm_fftMag_mfcc[5] numeric
@attribute pcm_fftMag_mfcc[6] numeric
@attribute pcm_fftMag_mfcc[7] numeric
@attribute pcm_fftMag_mfcc[8] numeric
@attribute pcm_fftMag_mfcc[9] numeric
@attribute pcm_fftMag_mfcc[10] numeric
@attribute pcm_fftMag_mfcc[11] numeric
@attribute pcm_fftMag_mfcc[12] numeric
@attribute pcm_LOGenergy numeric
@attribute class {No_Food,Apple,Haribo,Nectarine,Banana,Biscuit,Crisp}

@data

'Prob19_No_Food_1.wav',0.000000,-3.250689e+01,-1.639060e+01,6.267838e+00,-2.095726e+01,6.597322e+00,3.953855e+00,1.346384e+01,-5.285321e-01,-1.176270e+01,-8.163722e+00,8.926769e+00,3.862587e-02,1.483496e+01,No_Food
'Prob19_No_Food_1.wav',0.010000,-2.719806e+01,-1.719165e+01,-8.107703e+00,-2.403791e+01,-1.024285e+01,-6.333097e+00,1.794802e+00,-2.846059e-01,-1.214702e+01,-3.516420e+00,1.323622e+01,6.815407e+00,1.498715e+01,No_Food
'Prob19_No_Food_1.wav',0.020000,-2.677652e+01,-1.279993e+01,-6.967092e+00,-2.103788e+01,-3.286535e+00,2.327396e+00,-7.556728e+00,-3.875829e-01,-9.060074e+00,-2.073262e+01,8.452415e-01,1.351711e+01,1.446989e+01,No_Food
'Prob19_No_Food_1.wav',0.030000,-2.528762e+01,-1.063134e+01,-5.930324e+00,-1.638713e+01,-8.867039e+00,2.966618e+00,6.959305e+00,1.071679e+01,-3.571995e+00,-1.244267e+01,8.641685e+00,1.119380e+00,1.436013e+01,No_Food
'Prob19_No_Food_1.wav',0.040000,-1.477151e+01,-2.766893e+01,-1.502736e+01,-1.406891e+01,-1.289364e+01,-2.077906e+00,1.654333e+01,1.395391e+00,-8.914470e+00,-3.424679e+00,3.117570e+00,1.470896e+00,1.393601e+01,No_Food
'Prob19_No_Food_1.wav',0.050000,-1.564066e+01,-2.730897e+01,-9.942175e+00,-8.241878e+00,1.073270e+00,1.366332e-01,1.457072e+01,-2.474196e+00,-1.388130e+01,-1.365936e+01,-8.069144e+00,-5.311231e+00,1.364277e+01,No_Food
'Prob19_No_Food_1.wav',0.060000,-1.179908e+01,-2.057313e+01,-1.197151e+01,-1.343313e+01,-5.506250e+00,4.099829e+00,1.505171e+01,-4.550137e+00,-2.015779e+01,-9.103031e+00,-5.188558e+00,7.227353e-01,1.346854e+01,No_Food
'Prob19_No_Food_1.wav',0.070000,-1.267006e+01,-1.934192e+01,-1.651806e+01,9.526660e-01,-1.127300e+01,5.831690e+00,7.514180e+00,1.461225e+01,-1.849310e+01,-8.531794e+00,-2.905761e+00,1.612599e+01,1.347333e+01,No_Food
'Prob19_No_Food_1.wav',0.080000,-2.430949e+01,-1.193615e+01,-2.405663e+01,3.674203e-01,-2.474567e+01,2.052882e+01,7.623382e+00,1.611756e+01,-1.758513e+01,-7.395890e+00,-6.611455e+00,1.936058e+01,1.333734e+01,No_Food
'Prob19_No_Food_1.wav',0.090000,-1.984090e+01,-7.809028e+00,-1.727418e+01,-1.697966e+00,-1.198400e+01,1.315009e+01,5.627971e-01,1.012610e+01,-2.138927e+01,-1.703083e+01,-1.500324e+01,7.105649e+00,1.299803e+01,No_Food
'Prob19_No_Food_1.wav',0.100000,-1.832091e+01,-1.013803e+01,-2.563403e+00,-1.437446e+00,3.565968e+00,5.851185e+00,9.474144e-01,3.997355e+00,-1.127431e+01,-2.740545e+01,-1.871401e+01,1.355632e+01,1.204134e+01,No_Food
'Prob19_No_Food_1.wav',0.110000,-2.150358e+01,-1.632651e+00,2.952580e-01,-1.302603e+00,3.557243e-01,6.825325e+00,-2.308618e+00,-6.207929e+00,-7.436609e+00,-1.673846e+00,-1.921428e+01,6.543124e+00,1.239958e+01,No_Food
'Prob19_No_Food_1.wav',0.120000,-1.726532e+01,4.896791e+00,4.969835e-01,5.904603e+00,6.562255e+00,9.658154e+00,5.716319e+00,1.640971e+01,-7.481337e+00,-2.037669e+00,8.551467e+00,7.047046e+00,1.283244e+01,No_Food
'Prob19_No_Food_1.wav',0.130000,-1.875346e+01,1.834313e+00,-5.121733e-02,5.389313e-01,5.323496e+00,1.141359e+01,2.110998e+01,1.072849e+01,-1.082120e+01,-6.750721e+00,5.906575e+00,7.269865e+00,1.282197e+01,No_Food
'Prob19_No_Food_1.wav',0.140000,-2.166752e+01,3.804100e+00,4.988147e-01,-3.823110e+00,-4.630849e+00,7.585944e+00,8.929819e+00,4.414701e+00,-1.431046e+01,-6.118271e+00,-6.745447e+00,-7.318660e+00,1.193832e+01,No_Food
'Prob19_No_Food_1.wav',0.150000,-2.377972e+01,3.409528e+00,3.950238e+00,2.306728e+00,5.540388e+00,5.069889e+00,3.321918e+00,8.571683e+00,-3.679077e+00,8.062853e-01,-4.665273e+00,-4.904266e+00,1.130536e+01,No_Food
```

openXBOW tutorial

Universität Augsburg University

- A suitable MFCC based LLDS .arff has been supplied for the iHEARu-EAT data

```
openXBOWmaster\examples\tutorial\
```

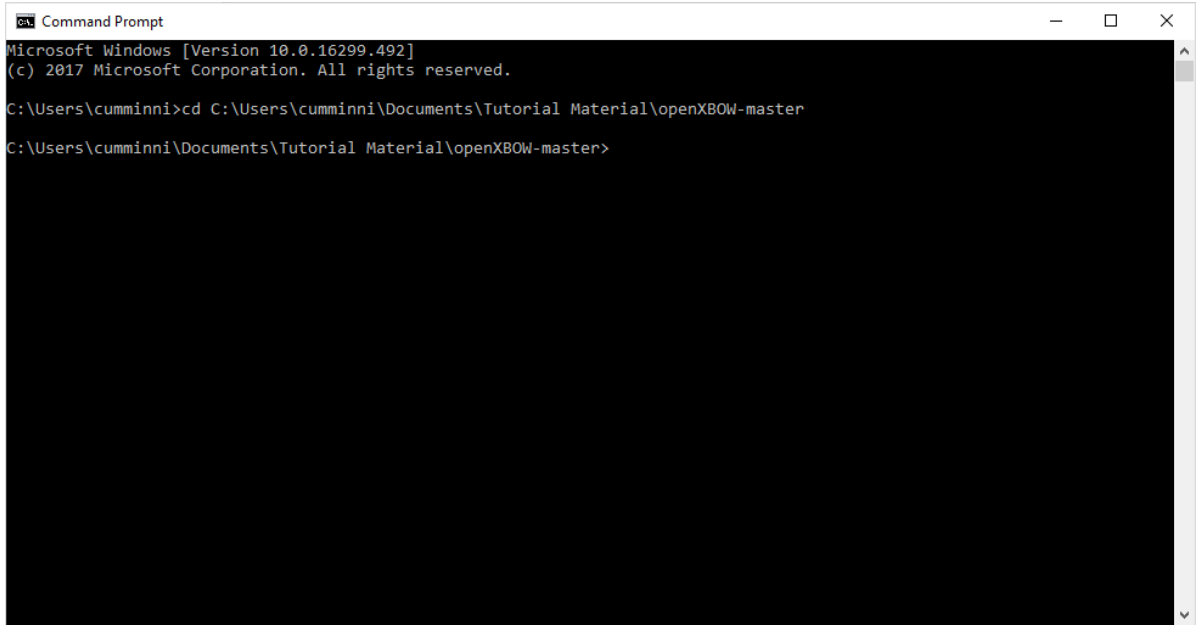
- Create output files in .arff format
- Subsequent classification can take place in WEKA

- Essential commands

```
-i p name/path to input file p
-o p name/path to input file p
-size p set size of codebook to p(default 500)
-a p assigns p vectors from codebook (default 1)
-c p method of creating codebook (default random)
-B p saves codebook p
-b p opens codebook p
```

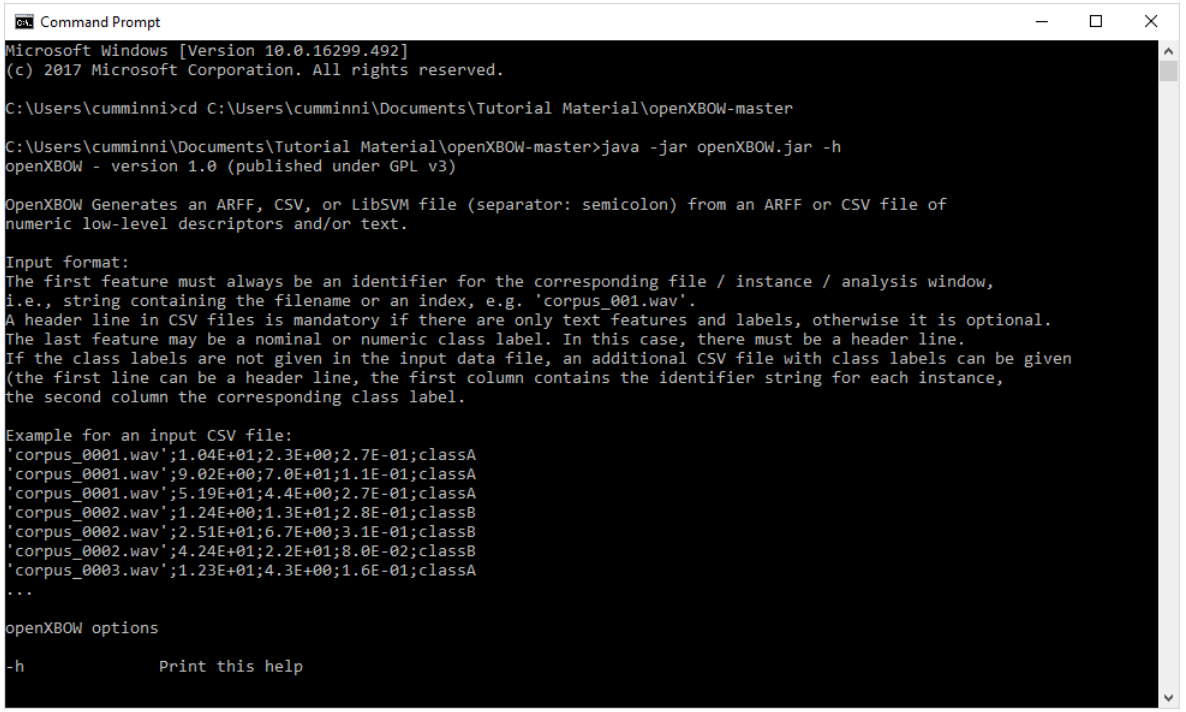
- Open windows command prompt
  - cd to folder containing openXBOW

```
C:\Users\cumminni\Documents\Tutorial Material\openXBOW-master
```



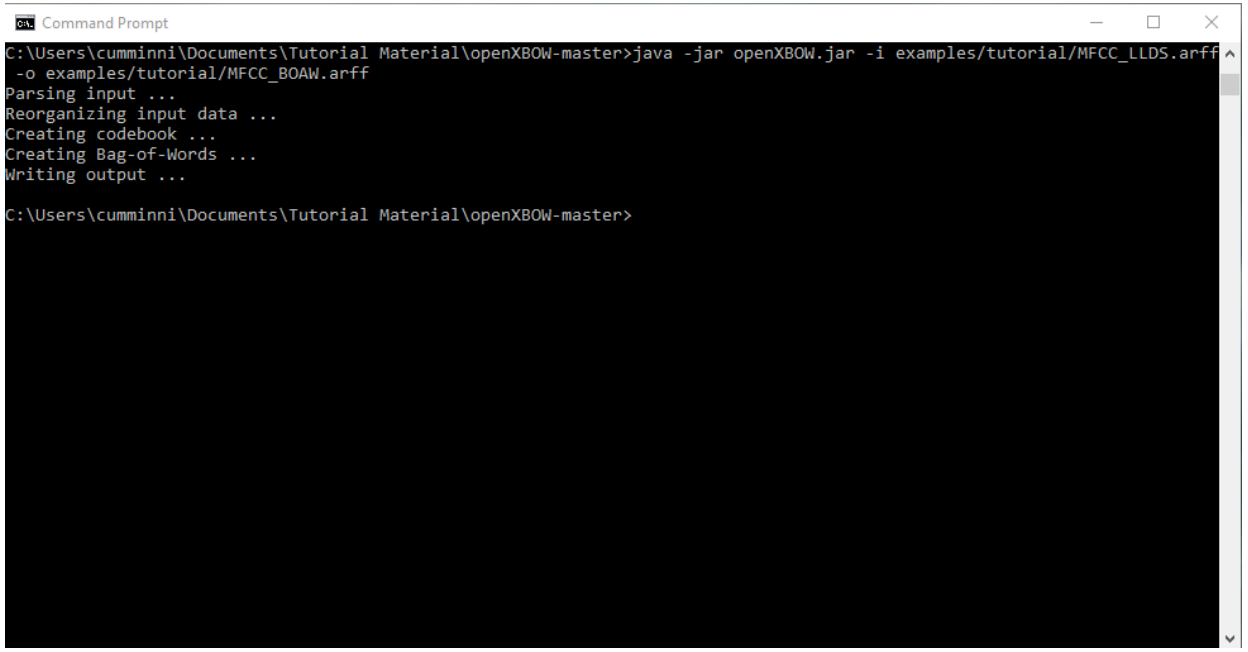
- Check openXBOW is working
  - call the instructions

```
>java -jar openXBOW.jar -h
```



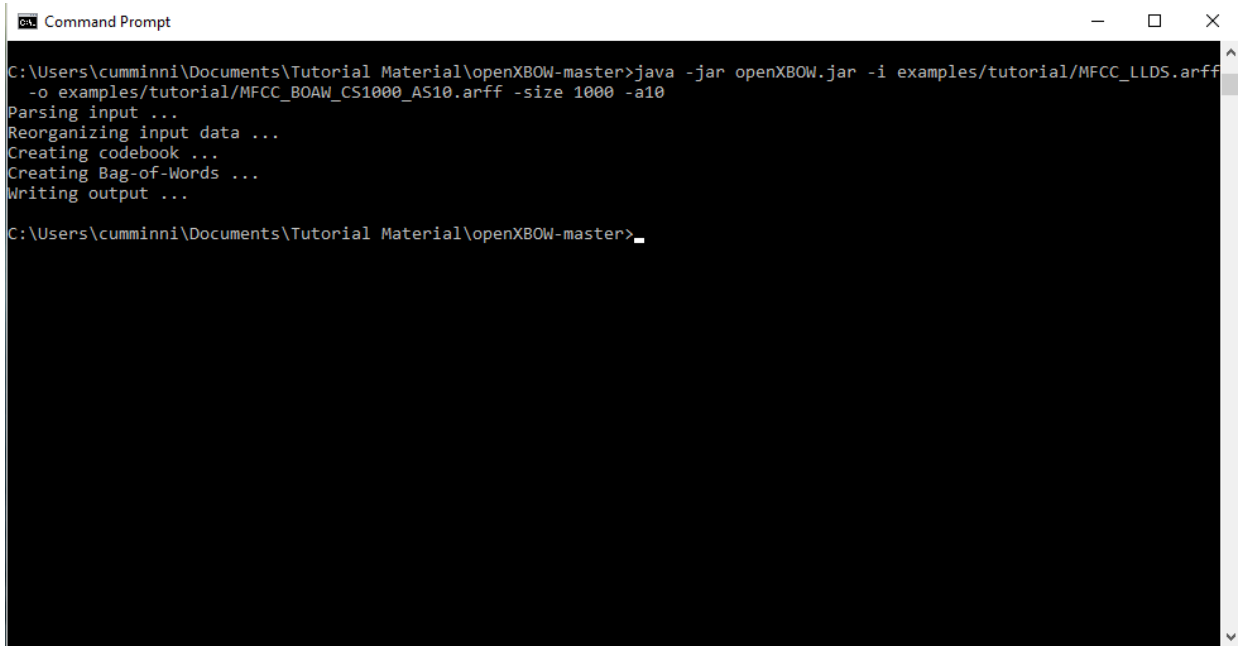
- Create a BoAW representation

```
java -jar openXBOW.jar
-i examples/tutorial/MFCC_LLDS.arff
-o examples/tutorial/MFCC_BOAW.arff
```



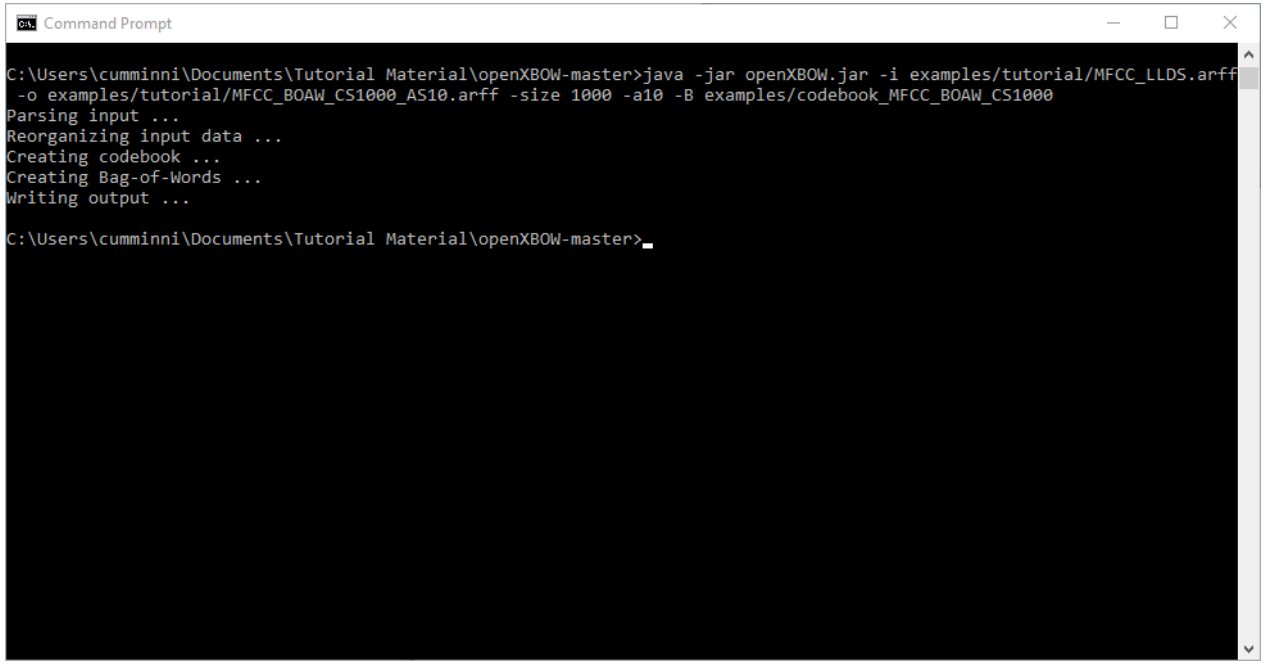
- Setting dictionary size (Cs = 1000) and assignments (Na = 10)

```
java -jar openXBOW.jar
-i examples/tutorial/MFCC_LLDS.arff
-o examples/tutorial/MFCC_BOAW_CS1000_AS10.arff
-size 1000 -a10
```



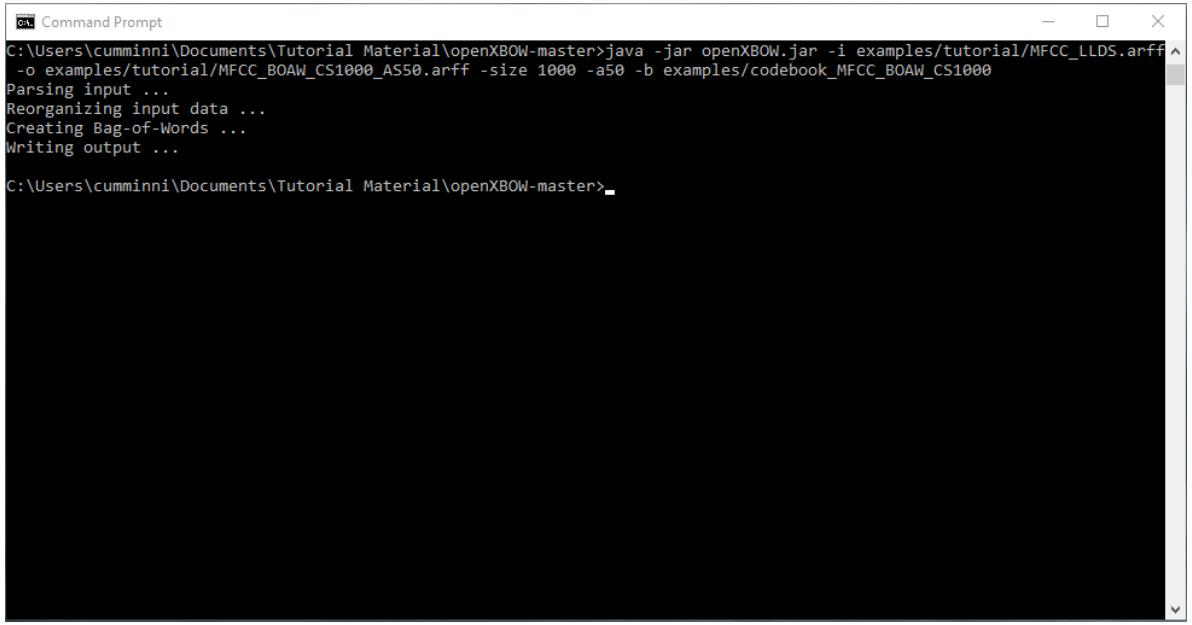
- Saving a codebook

```
java -jar openXBOW.jar
-i examples/tutorial/MFCC_LLDS.arff
-o examples/tutorial/MFCC_BOAW_CS1000_AS10.arff
-size 1000 -a10 -B examples/codebook_MFCC_BOAW_CS1000
```



- Loading a codebook

```
java -jar openXBOW.jar -i
examples/tutorial/MFCC_LLDS.arff -o
examples/tutorial/MFCC_BOAW_CS1000_AS50.arff -size
1000 -a50 -b examples/codebook_MFCC_BOAW_CS1000
```



- Other useful commands

```
-writeName Outputs the instance id in  the output file
-writeTimeStamp Output the time stamp in the output file

-standardizeInput Standardize all numeric input features
-normalizeInput Normalize all numeric input features

-gaussian p Soft assignment using Gaussian encoding with
standard deviation p

-log Logarithmic term weighting 'lg(TF+1)' of the term frequency
(compresses range of histogram)

-norm p Normalize the bag-of-features, 3 options:
       p=1: Divides the term frequencies (TF) by the number of
            input frames.
       p=2: Divides the TF by the sum of all TFs.
       p=3: Divides the TF by a factor so that the resulting
            Euclidean length is 1.
```

Contents

Universität Augsburg University

- CAS²T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- openXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

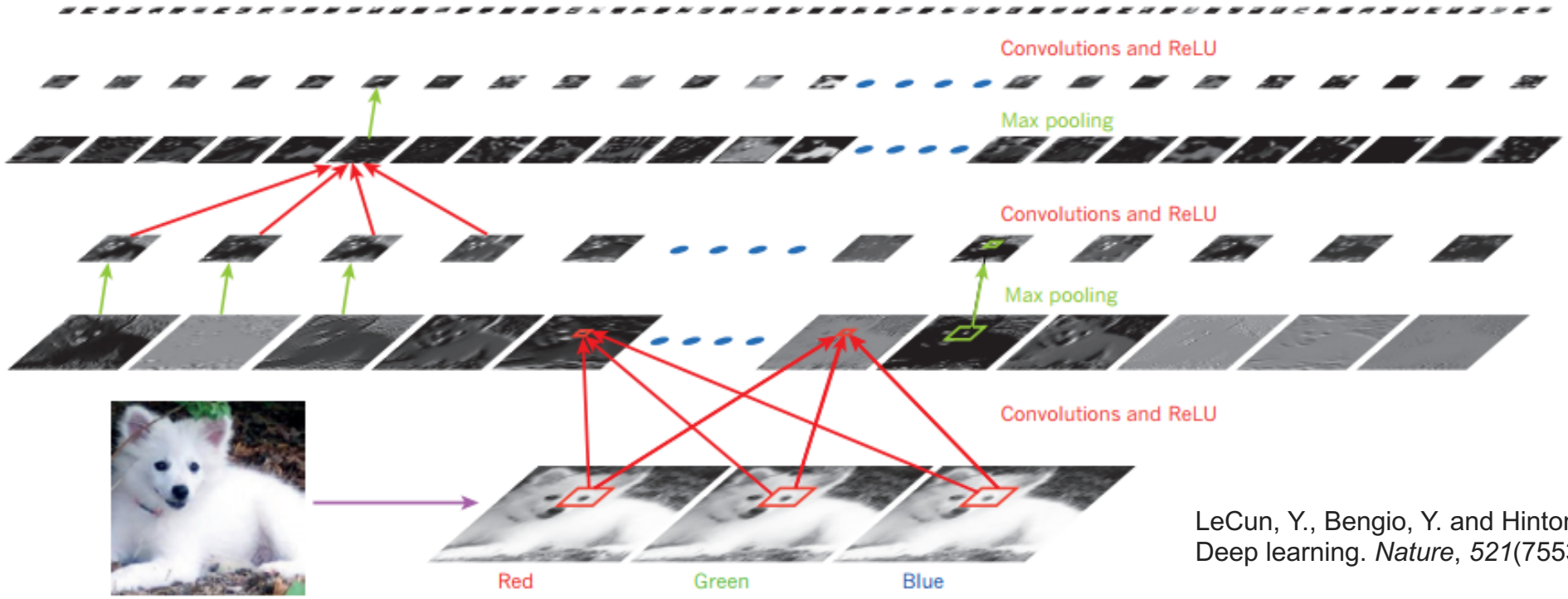
Feature Representation Learning

- Learning features directly from data
  - Conventional features are costly and labour intensive
    - Hand-crafted features may not capture suitable discriminative information for task at hand
  - Solution
    - Create learning algorithms that extract their own features
  - Widely used in conjunction with deep learning
    - Convolutional Neural Networks
      - Multiple layers of filtering

DeepSpectrum Features

Universität Augsburg University

- Feature Representation Learning
  - Widely used in image domain

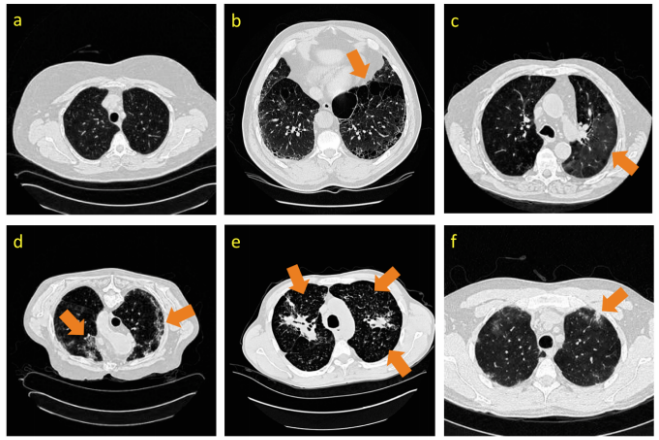


LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, *521*(7553), pp.436-444.

DeepSpectrum Features

Universität Augsburg University

- Example
  - Fine training image classification CNN's for interstitial lung disease (ILD) classification



(a): healthy; (b): emphysema; (c): ground glass; (d): fibrosis; (e): micronodules; (f): consolidation

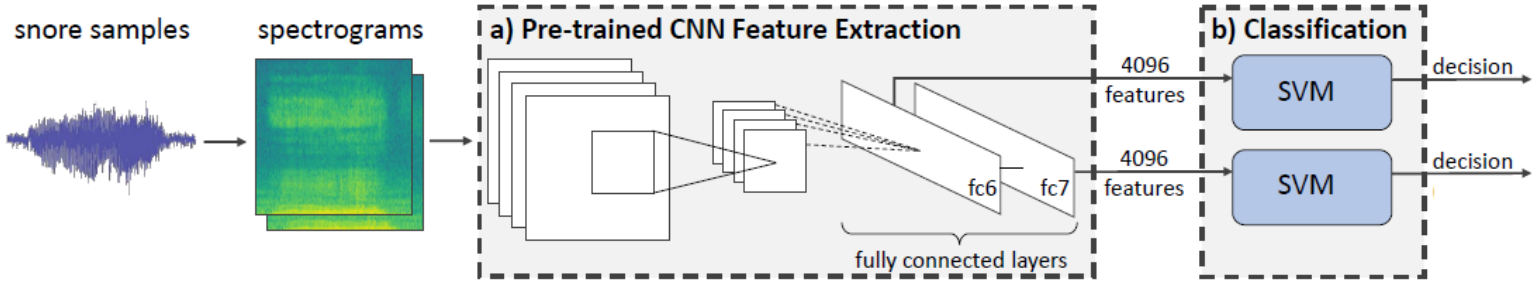
System	TPR
Purpose-Built	0.78
AlexNet	0.77
GoogLeNet	0.85

H. C. Shin *et al.*, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285-1298, May 2016.

DeepSpectrum Features

Universität Augsburg University

- Use of pre-trained *image* CNN's for audio tasks
  - Derived from forwarding spectrograms through ImageNets and using the activations from the last fully connected layers
    - AlexNet, GoogleNet, VG19

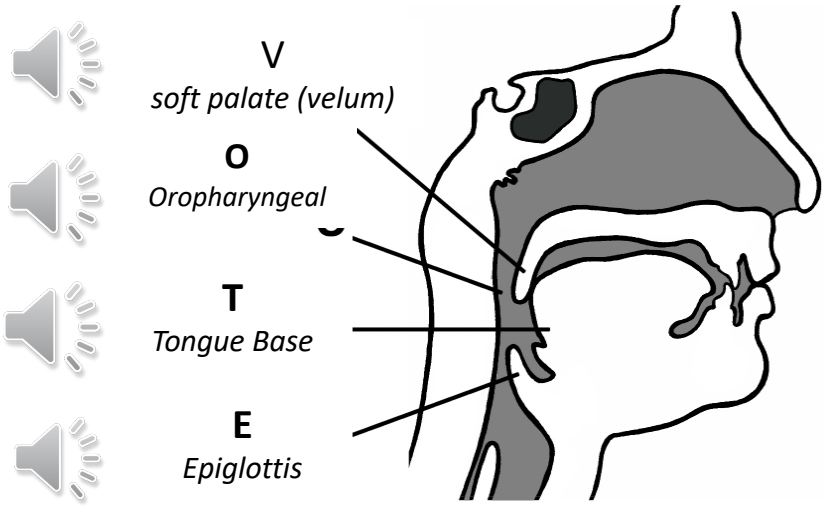


S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, and B. Schuller, "Snore Sound Classification Using Image-based Deep Spectrum Features," in *Proceedings INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*, (Stockholm, Sweden), pp 3512-3516 ISCA, August 2017.

DeepSpectrum Features

Universität Augsburg University

- Well suited to a range of speech & audio tasks
  - **4-Class Snore Sound Classification:**

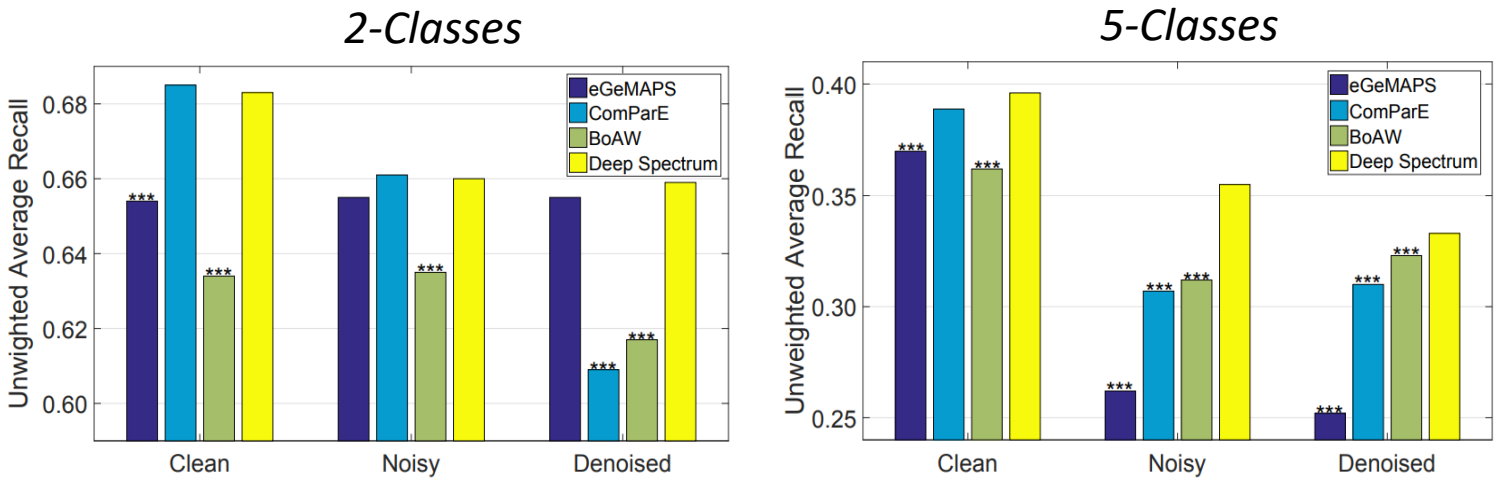


	UAR [%]	
Model	devel	test
Baseline End-2-End	40.3	40.3
Baseline Functional	40.6	58.5
Deep Spectrum	**44.8**	**67.0**

S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, and B. Schuller, “Snore Sound Classification Using Image-based Deep Spectrum Features,” in *Proceedings INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*, (Stockholm, Sweden), pp 3512-3516 ISCA, August 2017.

DeepSpectrum Features

- Well suited to a range of speech & audio tasks
  - **Speech-based emotion recognition**
    - FAU-AIBO dataset



N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. Schuller, "An Image-based Deep Spectrum Feature Representation for the Recognition of Emotional Speech," in Proceedings of the 25th ACM International Conference on Multimedia, MM 2017, (Mountain View, CA), pp 478-484, ACM, October 2017.

DeepSpectrum Toolkit

Universität Augsburg University

- Available at:
  - https://github.com/DeepSpectrum/DeepSpectrum
- Dependencies:
  - Python 3.6 with pipenv
    - For the Deep Spectrum tool (pip install pipenv)
  - Python 2.7
    - To download and convert the AlexNet model
      - Relies on `Caffe-tensorflow` conversion tool
      - https://github.com/ethereon/caffe-tensorflow

DeepSpectrum Toolkit

Universität Augsburg University

- **Installation**
  - Install the Deep Spectrum tool from the `deep-spectrum/` directory with `pipenv`

```
cd deep-spectrum
pipenv --site-packages install
```

- **Feature Extraction**
  - Simple command line arguments

```
pipenv run extract_ds_features -h
```

- Usage – other command line options
  - All options can also be displayed using `extract_ds_features -h`
- **Required options**

Option	Description	Default
-i	Specify the directory containing your *.wav* files or the path to a single *.wav* file.	None
-o	The location of the output feature file. Supported output formats are: • Comma separated value files and arff files. If the specified output file's extension is *.arff*, arff is chosen as format, otherwise the output will be in comma separated value format.	None

- **Extracting features from audio chunks**

Option	Description	Default
-t	Define window and hopsize for feature extraction. E.g -t 1 0.5 extracts features from 1 second chunks every 0.5 seconds.	Extract from the whole audio file.
-start	Set a start time (in seconds) from which features should be extracted from the audio files.	0
-end	Set an end time until which features should be extracted from the audio files.	None

- **Setting parameters for the audio plots**

Option	Description	Default
-mode	Type of plot to use in the system (Choose from: 'spectrogram', 'mel', 'chroma').	spectrogram
-scale	Scale for the y-axis of the plots used by the system (Choose from: 'linear', 'log' and 'mel'). This is ignored if mode=chroma or mode=mel. (default: linear)	
-ylim	Specify a limit for the y-axis in the spectrogram plot in frequency.	None
-delta	If specified, derivatives of the given order of the selected features are displayed in the plots used by the system.	None
-nmel	Number of melbands used for computing the melspectrogram. Only takes effect with mode=mel.	128
-nfft	The length of the FFT window used for creating the spectrograms in number of samples. Consider choosing smaller values when extracting from small segments.	The next power of two from 0.025 x sampling_rate _of_wav
-cmap	Choose a matplotlib colourmap for creating the spectrogram plots.	viridis

- **Parameters for the feature extractor CNN**

Option	Description	Default
net	Choose the net for feature extraction as specified in the config file	alexnet
-layer	Name of the layer from which features should be extracted as specified in your caffe .prototxt file.	fc7

- CNN Models supported in `Caffe-TensorFlow`
  - ResNet 152, ResNet 101 and ResNet 50
  - VGG 16
  - GoogLeNet
  - Network in Network
  - CaffeNet
  - AlexNet

- **Setting parameters for the audio plots**

Option	Description	Default
-mode	Type of plot to use in the system (Choose from: 'spectrogram', 'mel', 'chroma').	spectrogram
-scale	Scale for the y-axis of the plots used by the system (Choose from: 'linear', 'log' and 'mel'). This is ignored if mode=chroma or mode=mel. (default: linear)	
-ylim	Specify a limit for the y-axis in the spectrogram plot in frequency.	None
-delta	If specified, derivatives of the given order of the selected features are displayed in the plots used by the system.	None
-nmel	Number of melbands used for computing the melspectrogram. Only takes effect with mode=mel.	128
-nfft	The length of the FFT window used for creating the spectrograms in number of samples. Consider choosing smaller values when extracting from small segments.	The next power of two from 0.025 x sampling_rate _of_wav
-cmap	Choose a matplotlib colourmap for creating the spectrogram plots.	viridis

- **Defining label information**
  - Supports csv files for label information or explicitly set fixed labels for all input files

Option	Description	Default
-l	Specify a comma separated values file containing labels for each *.wav* file. It has to include a header and the first column must specify the name of the audio file (with extension!)	None
--tc	Set labeling of features to time continuous mode. Only works in conjunction with -t and the specified label file has to provide labels for the specified hops in its second column.	False
-el	Specify a single label that will be used for every input file explicitly.	None
--no_timestamps	Remove timestamps from the output.	Write timestamps in feature file.
--no_labels	Remove labels from the output.	Write labels in feature file.

Contents

Universität Augsburg University

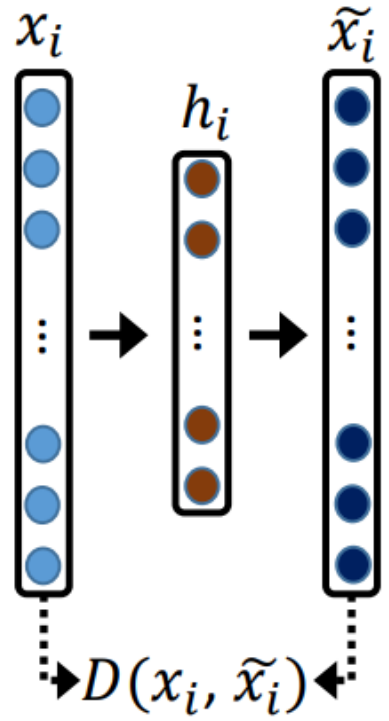
- CAS²T – Efficient Data Collection
- iHEARu-PLAY – Intelligent Data Annotation
- openSMILE – Feature extraction
- openXBOW – Multimodal Bag of Word generation
- DeepSpectrum – Image to Audio representations
- auDeep – Deep sequence-to-sequence autoencoder
- End2You – Multimodal End-to-End Learning toolkit

- Feature representation learning is difficult from sequential data
  - Deep neural networks typically require inputs of fixed dimensionality
    - E.g., Need to chunk audio for DeepSpectrum extraction
  - Alternative approach is to use sequence to sequence learning with *recurrent neural networks* (RNNs)
    - Enables the leaning of a fixed-length representations of variable-length sequences

Universität Augsburg University

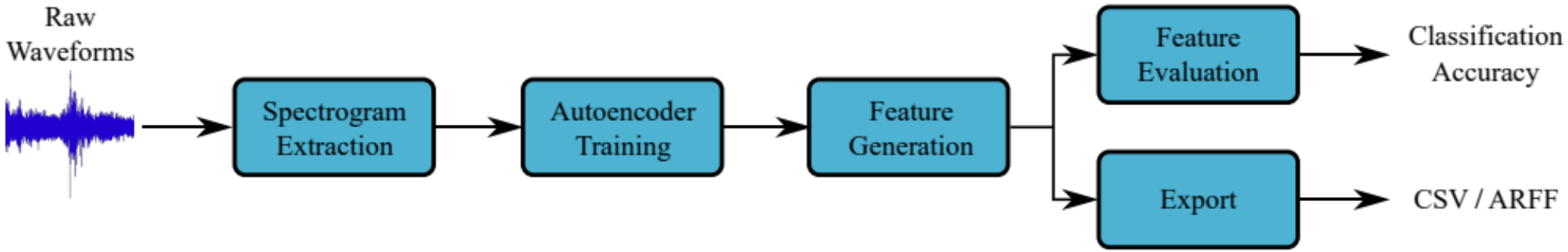
auDeep

- A Python toolkit for unsupervised feature learning
  - Feature extraction from audio data with deep recurrent autoencoders
  - Extensive command line interface
  - Python API



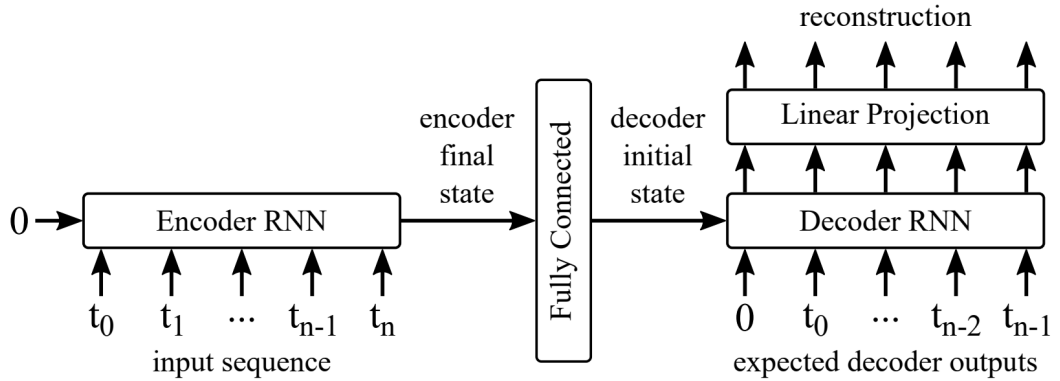
M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, “auDeep: Unsupervised learning of representations from audio with deep recurrent neural networks,” Journal of Machine Learning Research, vol. 19, 2018. 5 pages, to appear

- Representation learning is performed in five distinct stages
  1. Extraction of spectrograms from raw audio files
  2. Training of a DNN on the extracted spectrograms
  3. Feature generation using a trained DNN
  4. Evaluation of generated features
  5. Exporting generated features to CSV/ARFF

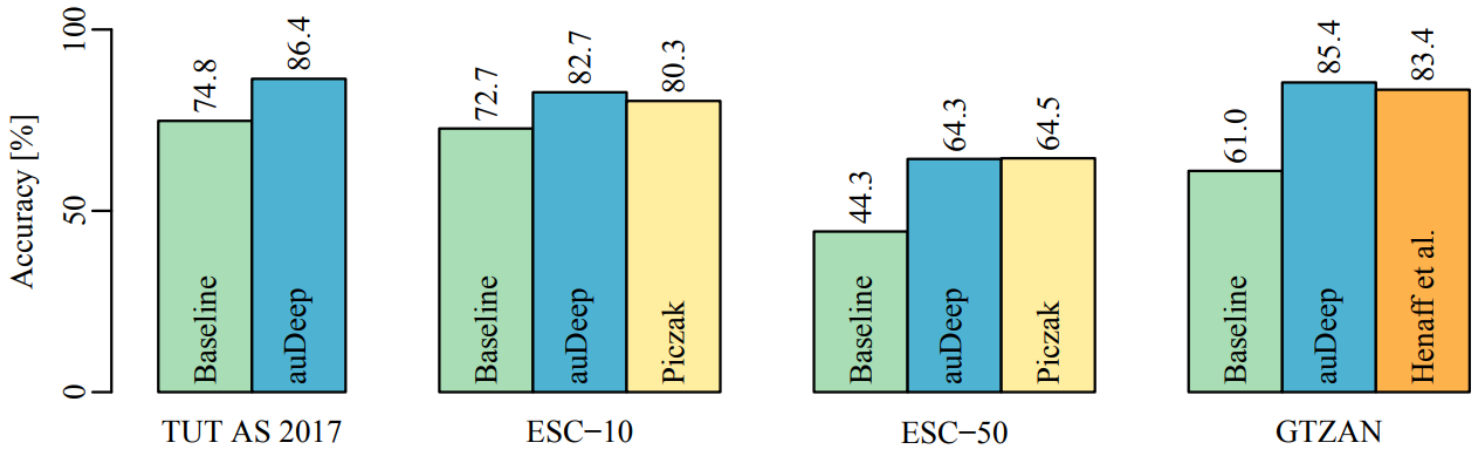


Deep Recurrent Autoencoders

1. Input sequences fed into an encoder RNN
2. Final hidden state of encoder transferred to a decoder RNN via a fully connected layer
3. Decoder RNN reconstructs the reversed input sequence
   – Introduces greater short-term dependencies between the encoder and the decoder



- Verified on a range of audio classification tasks
  - Acoustic scene classification (DCASE 2017)
  - Environmental Sound Classification (ESC-10 and ESC-50)
  - Music Genre Classification (GTZAN)
    - Outperforms baselines
    - Matches performance with state-of-the-art systems



auDeep Toolkit

Universität Augsburg University

- Available at:
  - https://github.com/auDeep/auDeep
- Dependencies:
  - Python 3.5
  - TkInter
  - Virtualenv
  - CUDA toolkit 8.0 (for GPU support)
  - cuDNN 5.1 (optional)
  - *Range of other Python dependencies installed during setup*

- **Installation**
  - Installation is through `pip` in a separate virtual environment

1. Start by creating a Python virtualenv for the installation

```
virtualenv -p python3 audeep_virtualenv
```

2. Subsequently, activate the virtualenv

```
Linux: > source audeep_virtualenv/bin/activate
Windows: > .\audeep_virtualenv\Scripts\activate.bat
```

3. Continue by installing auDeep

```
pip3 install ./auDeep
```

4. Finally the TensorFlow installation within the virtualenv needs to be patched

```
Linux: > patch audeep_virtualenv/lib/python3.5/site-packages/tensorflow/…
        python/framework/meta_graph.py auDeep/patches/fix_import_bug.patch

Windows: > \audeep_virtualenv\Lib\site-packages\tensorflow\python\...
           \framework\meta_graph.py .\auDeep\patches\fix_import_bug.patch
```

auDeep Toolkit

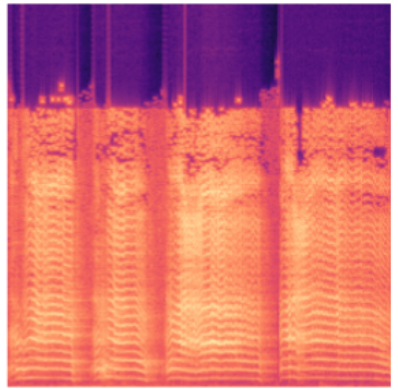
Universität Augsburg University

- Representation learning is performed in several distinct stages
  1. Extraction of spectrograms and metadata from raw audio files
     - `audeep preprocess`
  2. Training of a DNN on the extracted spectrograms
     - `audeep ... train`
  3. Feature generation using a trained DNN
     - `audeep ... generate`
  4. Evaluation of generated features
     - `audeep ... evaluate`
  5. Exporting generated features to CSV/ARFF
     - `audeep export`

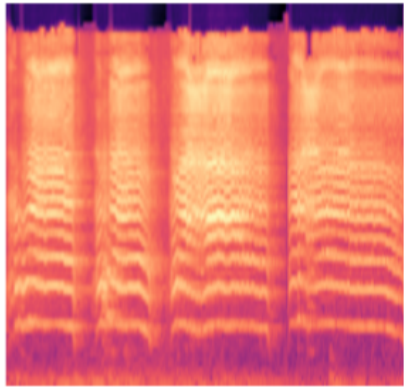
- Command line Interface
  - **Extracting Spectrograms**

```
$ audeep preprocess --options
```

  - Key options include:
    - FFT window width, FFT window overlap
    - Extraction of power spectra or mel-spectra
    - Chunking of audio file,
    - Amplitude clippings,
    - Handling of stereo data
      - Mean, left, right, difference



Spectrogram of a Baby Cry



Mel-Spectrum of a Baby Cry

- **Training Commands**

```
audeep ... train
```

  - Training progress can be monitored using Tensorboard.
  - Key options include
    - Minibatch size
    - Number of epochs
    - Learning rate
    - Number of layers
    - Number of units in each layer
    - Uni- or bi- directional
    - Corruption noise level

- **Evaluation and Prediction Commands**

```
audeep ... evaluat
```

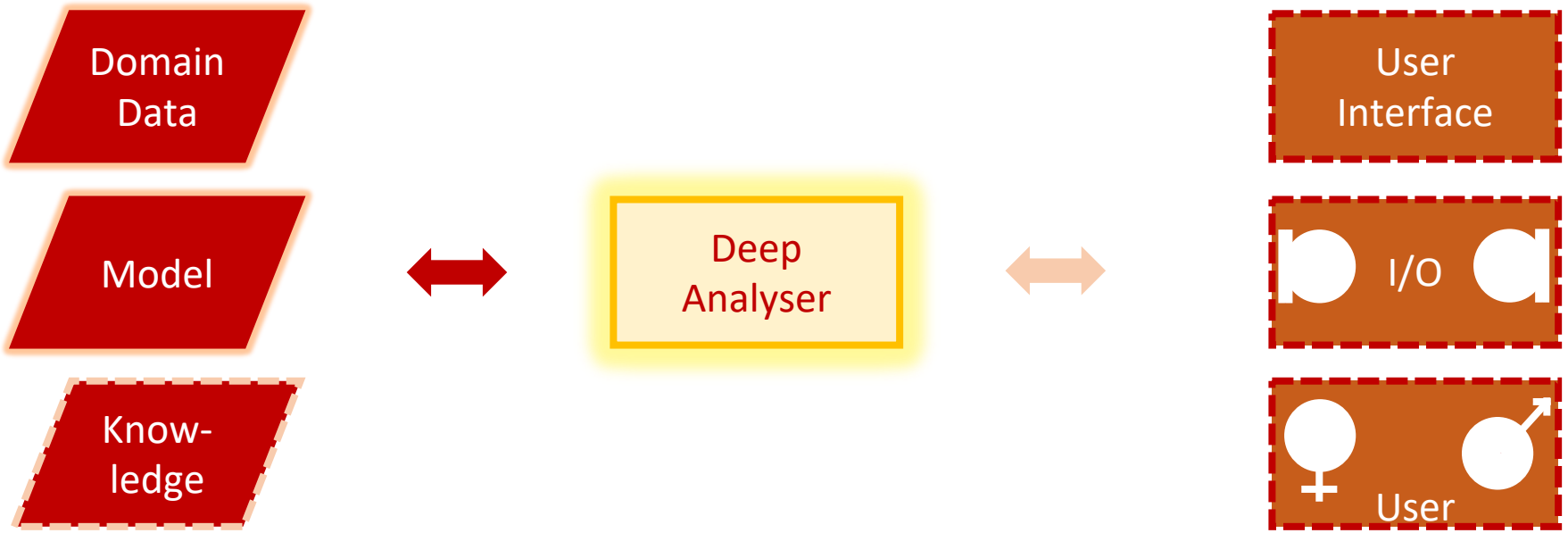
  - Key options include
    - Multilayer Perceptron or Support Vector Machine
      - With hyperparameter control
    - Partition based or cross-validation evaluation
    - Upsampling
    - Majority voting (with chunking)

- **Extensive Command line Interface**
  - An extensive range of other command line options
    - Inspection of input data
      - Number of files, number of labels, min and max lengths..
    - Parse raw input data
    - Import and export files in .arff or .csv format
    - Early and late fusion
  - Future version will include
    - Generative adversarial networks
    - Regressor options

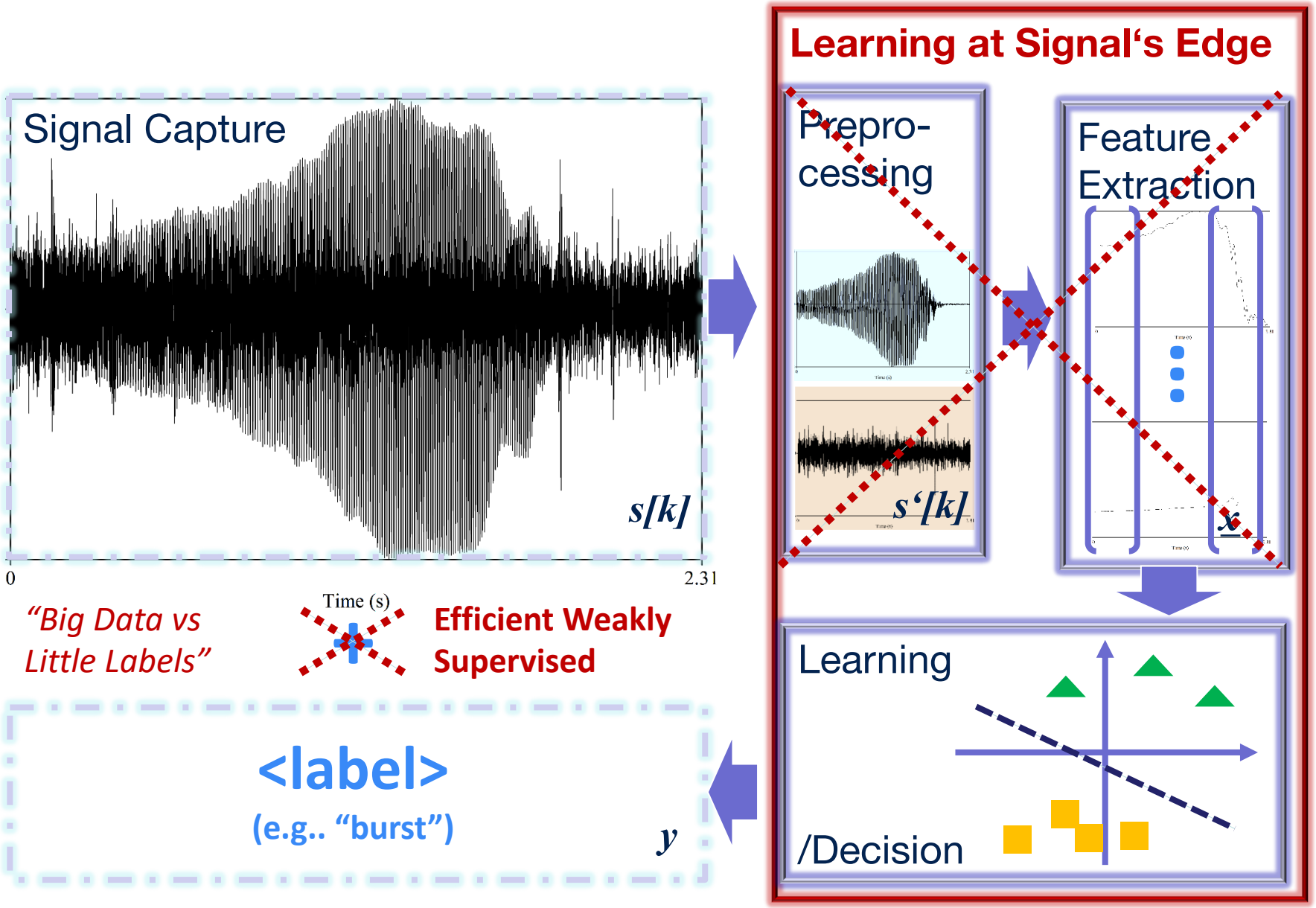
Contents

Pattern Recognition 2.0?

- The “Modern” Engine?



End-2-End Learning

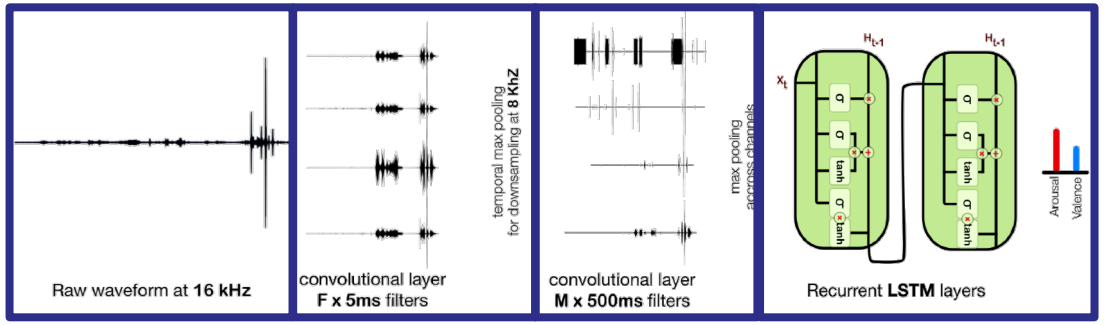


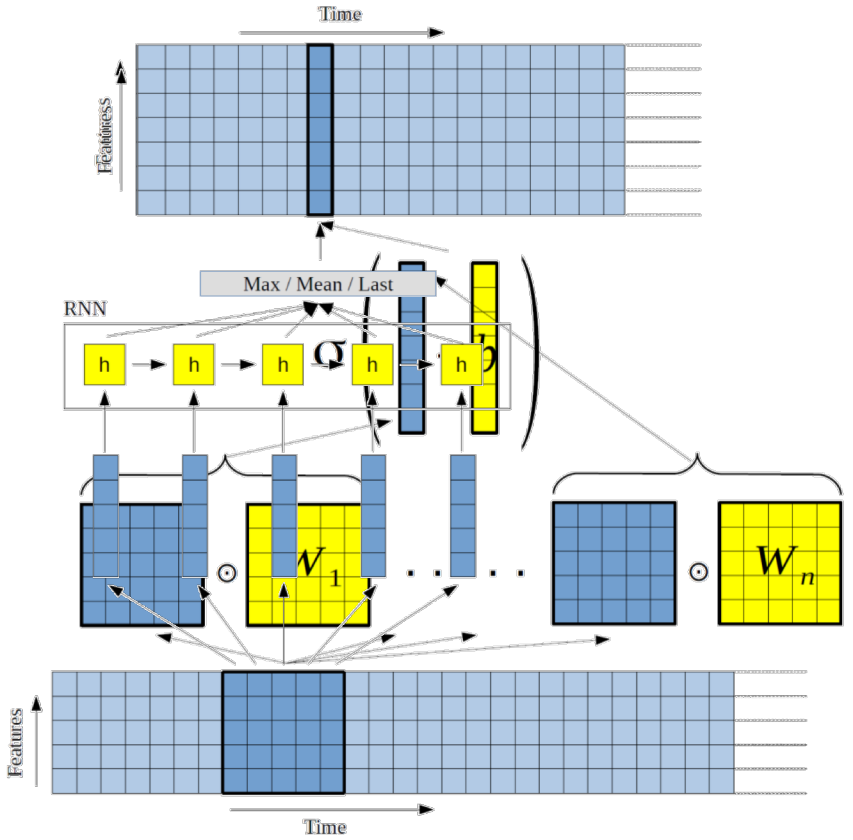
End-2-End Learning

Universität Augsburg University

- Pattern Recognition 2.0?
- CNN + LSTM → CLSTM ?

Arousal	CCC
Baseline	.366
e2e	.686

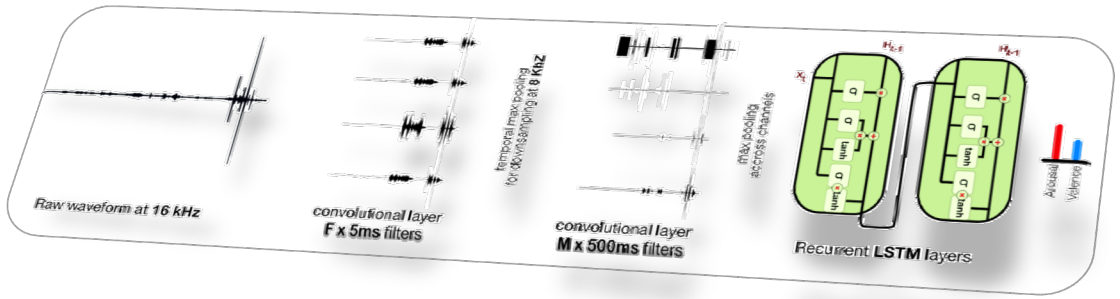


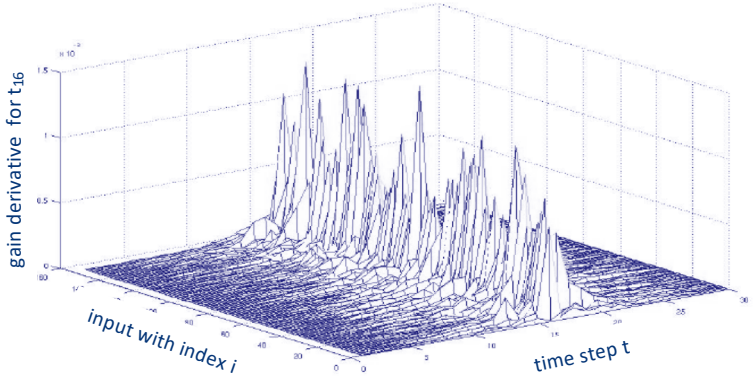


G. Trigeorgis, F. Ringeval, R. Bruckner, E. Marchi, M. Nicolaou, B. Schuller, and S. Zafeiriou, “Adieu Features? End-to-End Speech Emotion Recognition using a Deep Convolutional Recurrent Network,” in Proceedings 41st IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2016, (Shanghai, P. R. China), pp. 5200–5204, IEEE, IEEE, March 2016.

End-2-End Learning

- End-to-End – a black box?
  - CNN activations correlate with standard speech features

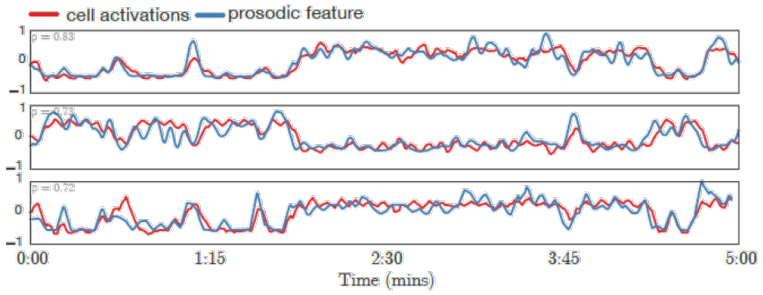




energy range (.77)

loudness (.73)

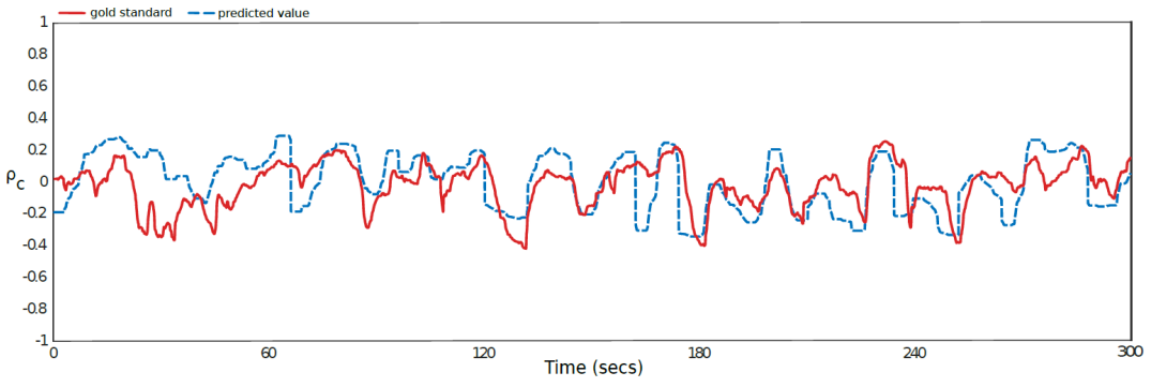
F0 mean (.71)



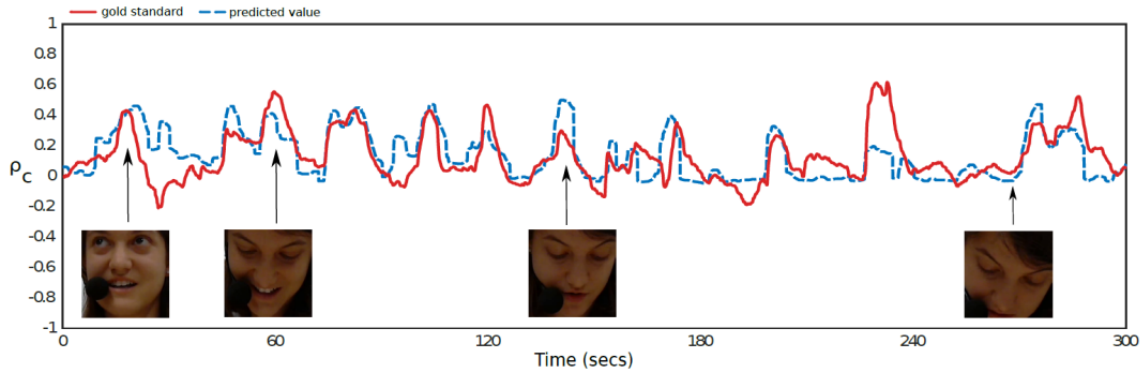
End-2-End Learning

- Adding Video

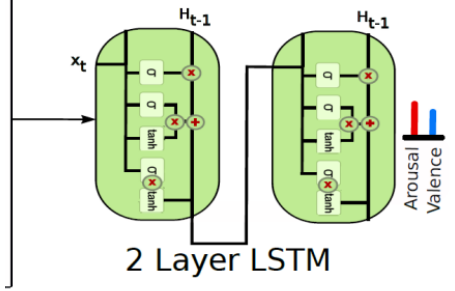
CCC	
Arousal	.789
Valence	.691



(a) Arousal



(b) Valence

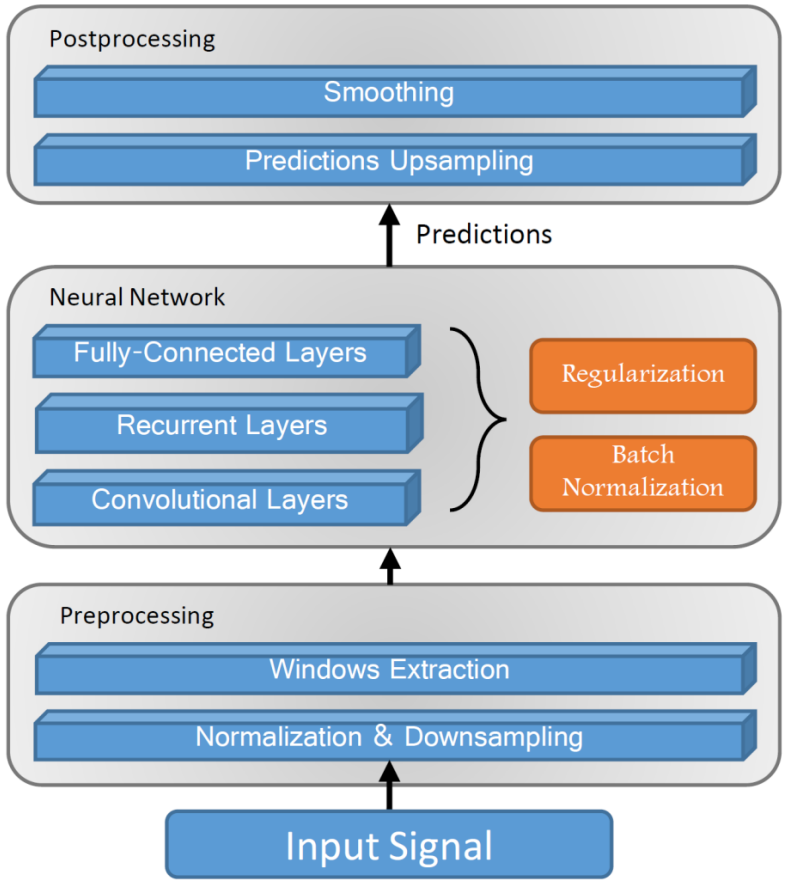


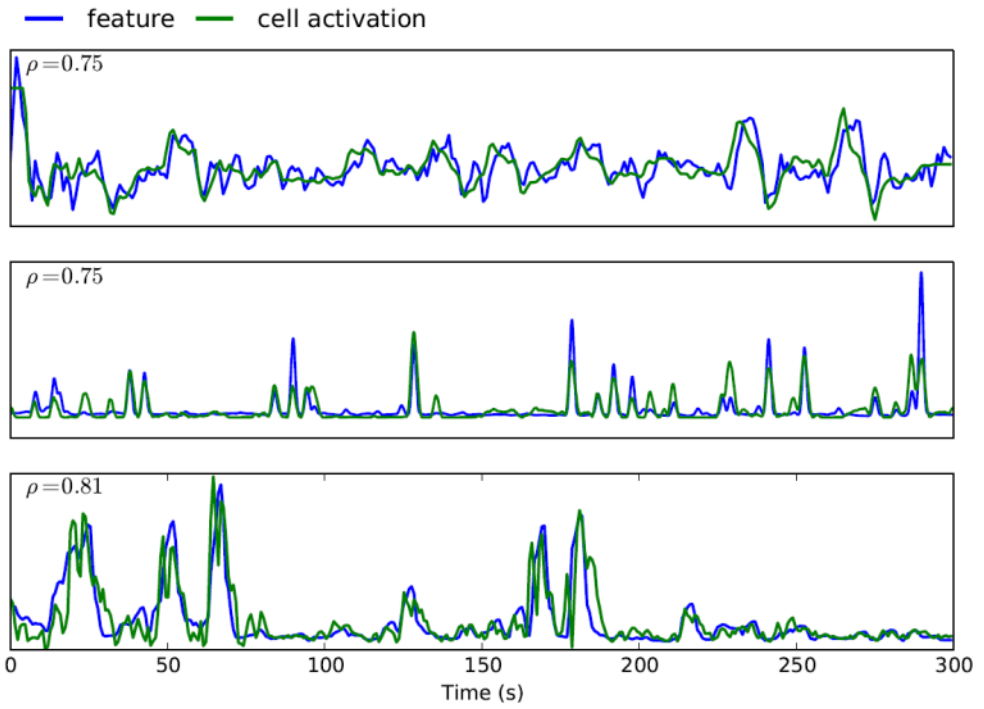
P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. W. Schuller and S. Zafeiriou, "End-to-End Multimodal Emotion Recognition Using Deep Neural Networks," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1301-1309, Dec. 2017.

End-2-End Learning

- Physiology

Fusion (Test)	CCC
Arousal	.430
Valence	.407

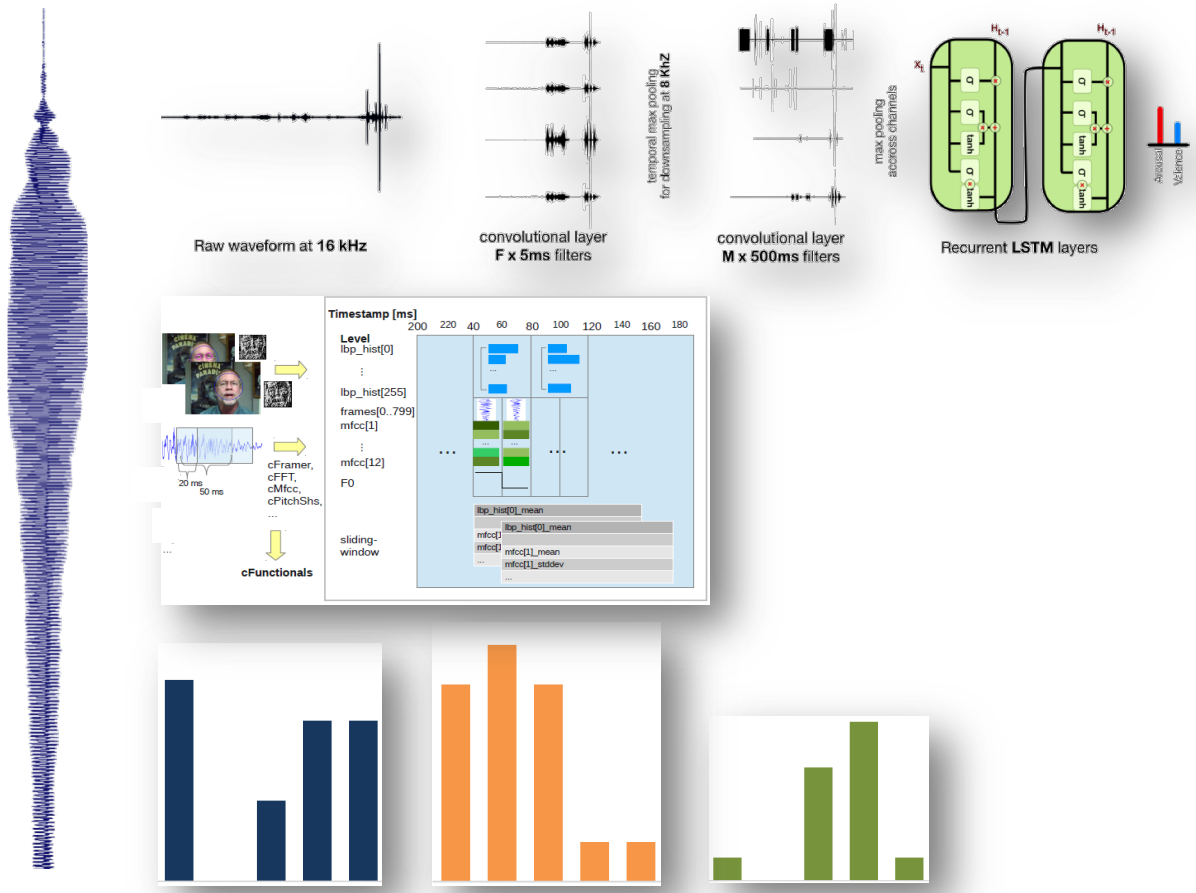




G. Keren, T. Kirschstein, E. Marchi, F. Ringeval, and B. Schuller, “End-to-end learning for dimensional emotionrecognition from physiological signals,” in *Proceedings 18th IEEE International Conference on Multimedia and Expo, ICME 2017*, (Hong Kong, P. R. China), pp IEEE, 985-990 July 2017.

End-2-End Learning

- e2e + functionals + BoAW?



Speech under Cold	%UA
func	70.2
BoAW	69.7
e2e	60.0
func + BoAW	70.1
e2e + func	64.8
e2e + BoAW	62.5
all (conf.)	70.7
all (maj. vote)	71.0

Universität Augsburg University

End2You - The Imperial Toolkit for Multimodal Profiling

- Available at: https://github.com/end2you/end2you
- Dependencies:
  - Python >= 3.4
  - NumPy >= 1.11.1
  - TensorFlow >= 1.4
  - MoviePy >= 0.2.2.11
  - liac-arff >= 2.0
  - sklearn >= 0.19

G. Trigeorgis, S. Zafeiriou, and B. Schuller “Adieu Features? End2You -- The Imperial Toolkit for Multimodal Profiling by End-to-End Learning,” arXiv preprint arXiv:1802.01115

- **Key Steps:**
  1. Install
  2. Generate the data
  3. Train the models
  4. Evaluate the models

1. Installation:

1. Download and install conda
2. Create a new conda environment and activate it:

```
$ conda create -n end2you python=3.5
$ source activate end2you
```

3. Install TensorFlow (v.1.4) following the official instructions and requirements
4. Clone and install the end2you project

```
(end2you)$ git clone git@github.com:end2you/end2you.git
```

2. Generate the data

- Convert the original files into a TensorFlow formant using TF Records
- User must supply a csv file that contains the full path of the raw data (e.g. .wav) and the label file for the data

```
file,label
/path/to/data/file1.wav,/path/to/labels/file1.csv
/path/to/data/file2.wav,/path/to/labels/file2.csv
```

- The label file should contain a column with the timestep and the later columns with the label(s) of the timestep

```
time,label1,label2
0.00,0.24,0.14
0.04,0.20,0.18
...
```

Universität Augsburg University

2. Generate the data

- To create the tfrecords you need to specify the flag to be `generate`

```
(end2you)$ python main.py --
tfrecords_folder=/where/to/save/tfrecords  \
                              --input_type=audio
                                generate  \
                              --data_file=data_file.csv \
```

- The `tfrecords` will be generated in a folder called `tf_records` which contains the converted files of the `data_file`

3. Train the models:

- Toolkit provides a default set of models
  - Audio: a 2-block of convolution max-pooling layers.
    1. A convolution layer of `40 filters` of `size 20` and a max-pooling layer of `size 2`. In the second block, the layer has
    2. A convolution layer `40 filters` of `size 40`, and pooling is applied to the feature maps with `size 10`
  - Video: A residual network (ResNet) with 50 layers* which has been widely used in the computer vision community.

* https://github.com/KaimingHe/deep-residual-networks
K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

3. Train the models:

- Toolkit provides a set of models
  - Recurrent Neural Network
    - Can be defined to be either a `GRU`- or a `LSTM`-type
      - Default is `GRU`
    - Flags can be set to control the `RNN` set-up:

Flag	Description	Values	Default
--hidden_units	The number of hidden units in the RNN.	int	128
--num_rnn_layers	The number of layers in the RNN model.	int	2
--seq_length	The sequence length to introduce to the RNN. If set to 0 indicates the whole raw file has a single label	int	150

3. Train the models:

- Training flags (other than RNN flags)

Flag	Description	Values	Default
--train_dir	Directory where to write checkpoints and event logs.	string	ckpt/train
--initial_learning_rate	Initial learning rate.	float	0.0001
--loss	Concordance Correlation Coefficient ('ccc') Mean Squared Error ('mse') Softmax Cross Entropy ('sce') Cross Entropy With Logits ('cewl')	ccc, mse, sce, cewl	cewl
--num_epochs	The number of epochs to run training.	int	50
--batch_size	The batch size to use.	int	2
--tfrecords_folder	The directory of the tfrecords files.	string	-
--tfrecords_eval_folder	If specified, after each epoch evaluation of the model is performed during training.	string	-

3. Train the models:

- Two different training processes:
  1. **Train only**
     - Will run for a user-defined number of epochs and performs only training.
     - To evaluate you need to separately run the `Evaluation` process
  2. **Train and evaluate model.**
     - The evaluation of the model is performed after each epoch,
       - The 5 best models are saved in the `--train_dir/top_k_models`
     - Stated with the `--tfrecords_eval_folder` flag
- **Example Training code:**

```
(end2you)$ python main.py --
tfrecords_folder=path/to/tfrecords \
                              --input_type=audio \
                              train \
                              --train_dir=ckpt/train \
```

4. Evaluate the model

- *Perfomed when the `--tfrecords_eval_folder` flag is not set in training*
- List of flags can be set for evaluation:

Flag	Description	Values	Default
--train_dir	Directory where to write checkpoints and event logs.	string	ckpt/train
--log_dir	Directory where to write event logs.	float	0.0001
--metric	Which metric to use for evaluation. One of: Concordance Correlation Coefficient (ccc) Mean Squared Error (mse) Unweighted Average Recall (uar)	ccc, mse, uar	'uar'
--eval_interval_secs	How often to run the evaluation (in sec).	int	300

4. Evaluate the model

- Example evaluation code:

```
(end2you)$ python main.py --
tfrecords_folder=path/to/tfrecords \
                              --input_type=audio \
                                evaluate \
                              --train_dir=ckpt/train \
                              --log_dir=ckpt/log
```

- Evaluation can also be performed through `TensorBoard`

```
(end2you)$ tensorboard --logdir=ckpt
```