INF642 – Socio-emotional Embodied Conversational Agents

Lab 1: Acoustic and Visual Data Preprocessing

Martín Cepeda

January 20, 2021

This report explains the steps to compute prosodic and visual features extracted from different videos. The videos are a selection of TED Talks, which are mostly focused on a single person speaking about a particular subject during at least 10 minutes, thus exhibiting different sentiments both in the person's tone and facial expressions during its speech.

Attached to this report there's the file `TD1.zip` which has a Github repository structure of the scripts developed. A `README` file is provided to replicate the presented results. As the Windows binaries were available, no compilation was needed for OpenSmile nor OpenFace.

Feature sets are available here.

1 Getting the data

As suggested by the lab instructions, the openly available tool `youtube-dl` was used to download TED Talks from YouTube, via the following template call:

```
youtube-dl -o ".\data\%(id)s.%(ext)s" --batch-file ".\VideoIDs.txt" -f mp4
```

Which downloads all videos listed in `VideoIDs.txt` and saves them in mp4 format using their unique ID as filename to the `data` directory.

Using PowerShell built-in functions, the library `ffmpeg` is used to extract the audio from the previously downloaded videos, using a 44.1KHz sampling using 2 channels and the `pcm_s16le` codec for muxing into WAV :

```
foreach ($i in @(gci ".\data" -file *.mp4)){ \
ffmpeg -i ".\data\$i" -vn -acodec pcm_s16le -ar 44100 -ac 2 ".\data\audio\$i.wav"}
```

An additional command is executed to rename the files and thus keep an `ID.ext` convention.

2 Computing features

2.1 Video

For video features, a pre-trained OpenFace model [1], which uses several neural network approaches [2], [3], [4], [5] to detect and track facial landmarks, eye gaze and facial Action Units. This is done by a call like the following:

```
foreach ($i in @(gci ".\data" -file *.mp4)){ \
FeatureExtraction -root .\data -f $i -out_dir .\data\face_features}
```

This generates in the `data\face_features` directory the following files for each video:

- Inside `ID_aligned`: a set of 112x112 images that contain (when found) the face identified along all the frames of the video.
- `ID.avi`: a no-audio video of the face landmarks, eye gaze and face orientation drawn over the input video.

- `ID.hog`: binary file containing histogram of oriented gradients (HOG) features over all frames of the video.

- `ID.csv`: per-frame summary of features (drawn in the AVI file) with an associated confidence and success flag

2.2 Audio features

OpenSmile can compute an extensive set of audio features, and the computation works based on a modular approach specified in a configuration file. In brief words, one must specify a graph of operations and parameters with built-in signal analysis modules that perform the requested operations. For instance, in order to compute Cepstral coefficients the sequence of operations to apply to an input WAV file is the following:

$$read-file \rightarrow divide-in-frames \rightarrow apply-window-function \rightarrow fft \rightarrow compute-magnitude \rightarrow mfcc$$

This same idea is replicated in the global configuration file, which was written from scratch using as reference the available config files in the documentation. The operation graph of submitted config file is the following:

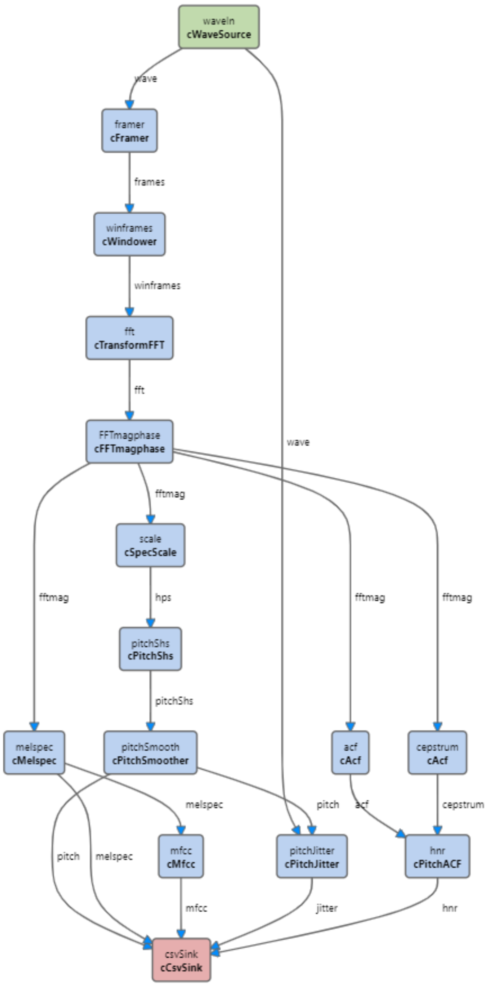


Figure 1: This visualization was generated using the openSMILE Config Files Visual Studio Code extension.

With this configuration file is possible to extract, among others, the following features:

- F0 and F0 envelope

- Mel Frequency coefficients (first 14)
- Jitter, Shimmer and Shimmer envelope
- HNR

By calling:

```
foreach ($i in @(gci ".\data\audio" -file *.wav)){ \
SMILExtract -C ".\src\MyConf.conf" -I ".\data\audio\$i" -O ".\data\audio_features\$i.csv"}
```

The full configuration file is available in the submission ZIP.

2.3 Making a common audio-video dataframe

Due to the difference of sampling rates for video and audio features, the former given by the variable FPS used in video compression and the latter given by the "framer" module parameters, the common database was made after the following steps:

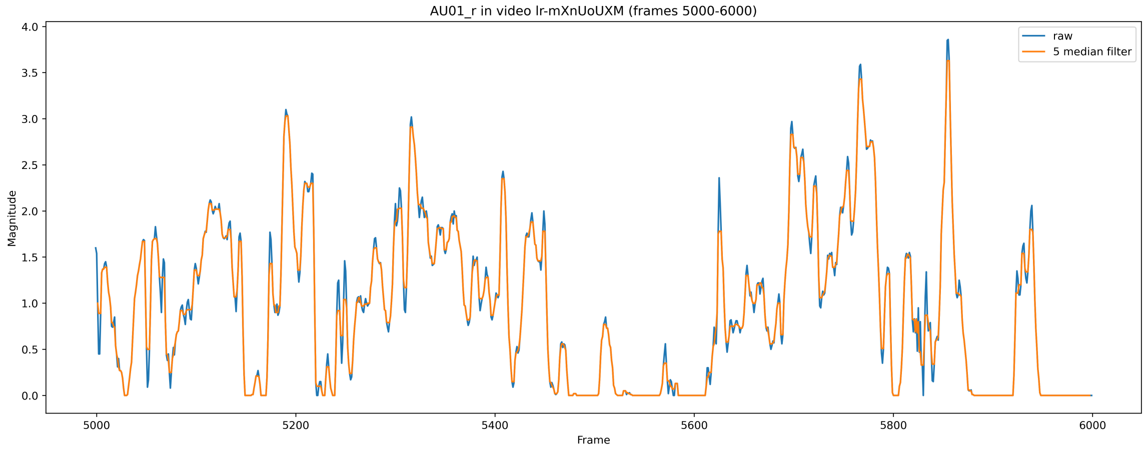
1. Drop frame number columns in audio dataframe (irrelevant)
2. Merge both dataframes on the timestamp: merged dataframe has the union of timestamps in audio and video features.
3. Back-fill missing values (as the timestamp i indicate the end of frame i)
4. Drop rows that correspond to a failure in recognizing face features. Because of the previous step, this captures audio frames also.
5. Drop frame number and success flag columns (irrelevant after previous filter)

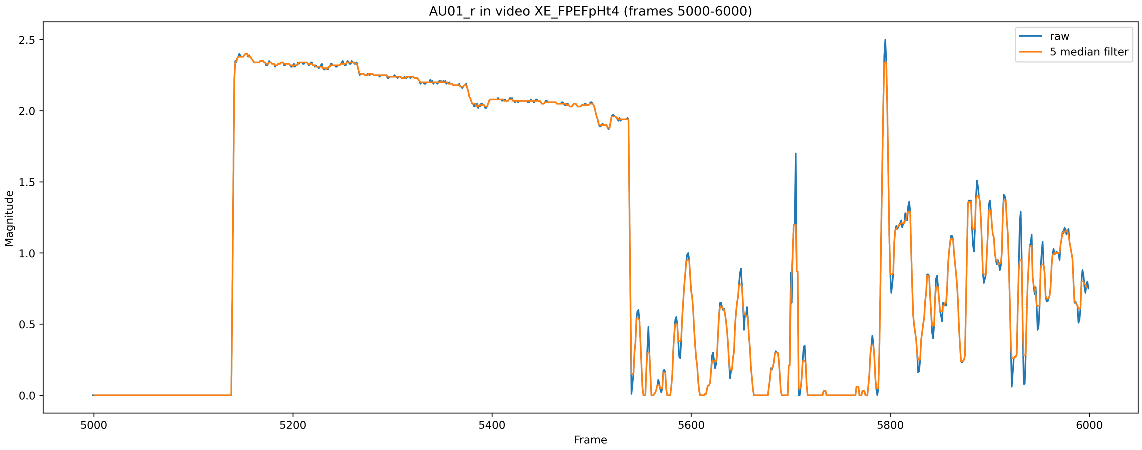
The resulting common feature collection for each video was saved to CSV and they are available in the link mentioned at the beginning of this report. Also the Python notebook used to generate the merged datasets and visualizations is in the ZIP file attached.

3 (very) Exploratory visualizations

3.1 Video features

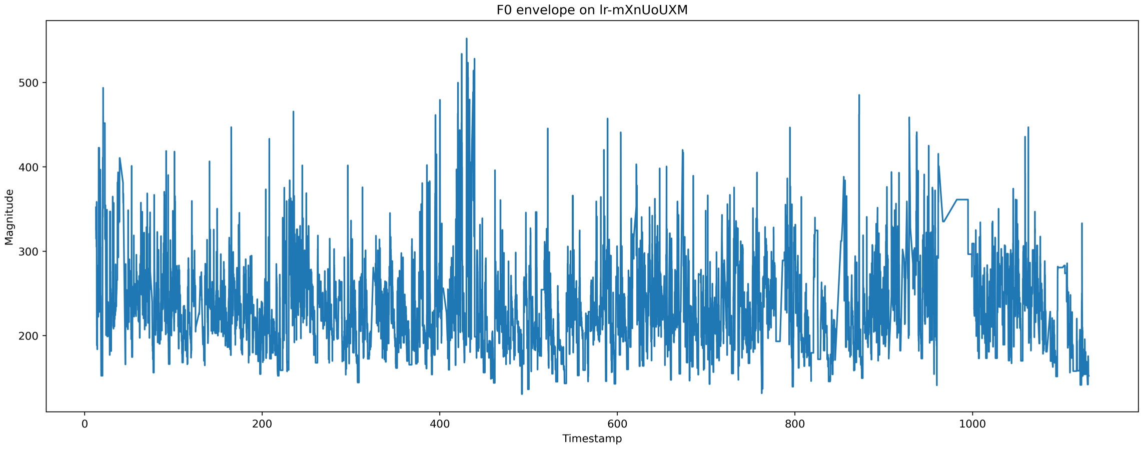
The following features were recovered from the "raw" frames (no filtering of unsuccessful feature recognition):

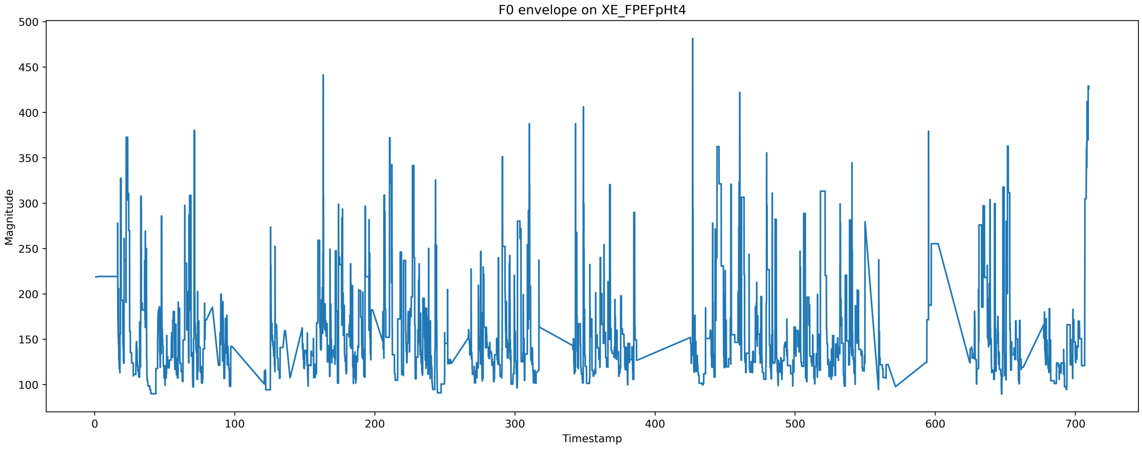




3.2 Audio features

The following features were recovered from filtered frames (timestamps) of the same video IDs as the video features, but along all timestamps:





References

[1] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Openface: An open source facial behavior analysis toolkit," *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–10, 2016.

[2] A. Zadeh, T. Baltrušaitis, and L.-P. Morency, "Convolutional experts constrained local model for facial landmark detection," 2017.

[3] T. Baltrusaitis, P. Robinson, and L. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 354–361.

[4] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling, "Rendering of eyes for eye-shape registration and gaze estimation," 2015.

[5] T. Baltrušaitis, M. Mahmoud, and P. Robinson, "Cross-dataset learning and person-specific normalisation for automatic action unit detection," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 06, 2015, pp. 1–6.