eRS - A System to Facilitate Emotion Recognition in Movies

Joël Dumoulin, Diana Affi, Elena Mugellini, Omar Abou Khaled
HumanTech Institute, University of Applied Sciences
Fribourg, Switzerland
[name.surname]@hes-so.ch

SUBMITTED to ACM MULTIMEDIA 2015 OPEN SOURCE SOFTWARE COMPETITION

ABSTRACT

We present e*RS*, an open-source system whose purpose is to facilitate the workflow of emotion recognition in movies, released under the MIT license. The system consists of a Django project and an AngularJS web application. It allows to easily create emotional video datasets, process the videos, extract the features and model the emotion. All data is exposed by a REST API, making it available not only to the e*RS* web application, but also to other applications. All visualizations are interactive and linked to the playing video, allowing researchers to easily analyze the results of their algorithms. The system currently runs on Linux and OS X. e*RS* can be extended, to integrate new features and algorithms needed in the different steps of emotion recognition in movies.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms; Design; Experimentation

Keywords

Open Source; Sentiment Analysis; Video Analysis

1. INTRODUCTION

In recent years, there has been a growing interest in research on multimedia retrieval based on concepts such as emotion. Proposed by Hanjalic *et al.* [6], *affective content analysis* is a well used approach that suggests to extract and model the affective content of a video (using both audio and video features) corresponding to the emotions (type and intensity) that are expected to arise in the user while watching the video. For researchers, the workflow to apply this kind of approach in order to model the emotion present in movies usually consists of the following steps. First, datasets of videos are created, corresponding to a specific video database or a specific experiment objective. Then, videos are processed in order to extract features. Finally, the features are used to model the emotion. If many great libraries and toolboxes are available for processing audio and video files (i.e., OpenCV [1], OpenSMILE [3]), using these libraries, managing the files and the extracted data, analyzing and comparing results can be a tedious task. Particularly when it comes to analyzing results, trying to map a static two dimensional plot with a dynamic playing video.

The goal of the e*RS* system is to facilitate this workflow (illustrated in Figure 1) for researchers, providing them with a backend handling of the processing tasks, and a web application allowing to easily manage all the data and the tasks, while helping them as well to analyze results thanks to interactive visualizations.

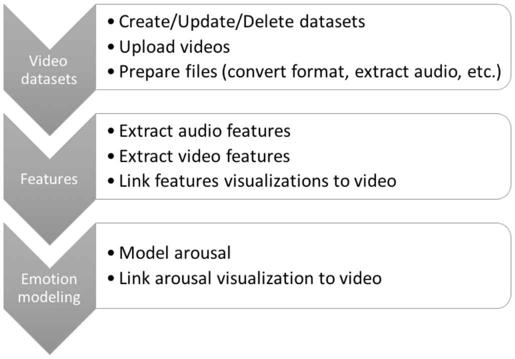

Video datasets
• Create/Update/Delete datasets
• Upload videos
• Prepare files (convert format, extract audio, etc.)
Features
• Extract audio features
• Extract video features
• Link features visualizations to video
Emotion modeling
• Model arousal
• Link arousal visualization to video

Figure 1: System workflow overview - 1) Video datasets management 2) Features extraction and visualization 3) Emotion modeling

2. OVERALL ARCHITECTURE

The e*RS* system consists of two main parts as shown in Figure 2: the backend and the frontend.

2.1 Backend

The backend is expected to be run on a server, allowing to take advantage of high computational capabilities, required by tasks such as video conversions, features extraction and

MM'15, October 26–30, 2015, Brisbane, Australia.

ACM 978-1-4503-3459-4/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2733373.2807409 .

emotion modeling. It consists of a Django[1] project and is therefore written in Python. For each dataset, all videos files are stored in folders according to their emotional class. All datasets and videos metadata are stored in a relational database. The default relational database management system used is MySQL, but it can be replaced by any other Django supported database type (i.e., PostgreSQL, Oracle).

The time consuming tasks are launched as Celery[2] asynchronous tasks (Celery is a distributed task queue library), using a Redis[3] message broker.

2.2 Frontend

The frontend is an HTML5 web application, developed with the AngularJS[4] JavaScript framework. The role of this web application is to offer a pleasant and easy to use interface to the system users, allowing to launch tasks on the backend and retrieve data for analysis, with the use of interactive visualizations.

It can have access to the data by two different ways: through a Representational State Transfer (REST) API and through a WebSocket endpoint. The choice of the REST API is motivated by the idea that other applications than the e*RS* web application could have access to the data and take advantage of it. The use of the WebSocket[5] endpoint allows to provide monitoring capabilities, by establishing a full duplex communication between the frontend and the backend. It is therefore possible to launch time consuming tasks in an asynchronous manner, close the browser, and when the web application will be launched the next time it will be updated with the current state of the tasks, thanks to the push communication model.

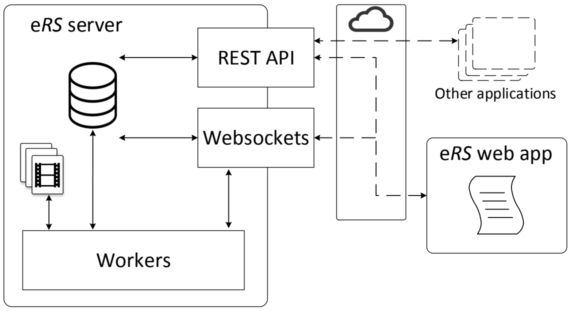


Figure 2: Overall architecture - the server stores the videos and all the associated data ; as the extracted data (datasets and video metadata, features, emotion values, etc.) are available through a REST API, the e*RS* web application but also other applications can access them ; the e*RS* web application launches processing tasks asynchronously and monitors them through the WebSockets endpoint.

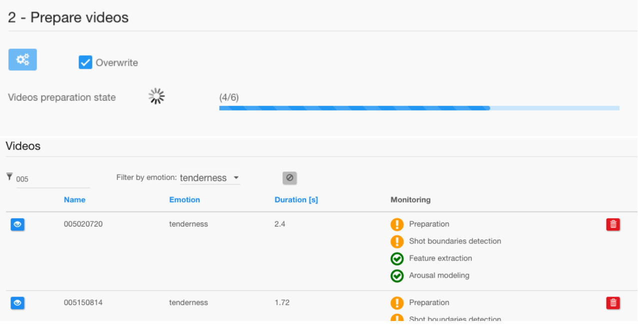
3. ERS FUNCTIONALITIES

In this section, an overview of the functionalities provided by the e*RS* system are described.

3.1 Video datasets management

Video datasets can be created, updated and removed. Videos can be uploaded to the server directly through the web application. It is also possible to add the files in the correct folders on the server and ask the application to scan for available videos. Once added to the system, videos have to be converted in a format so they can be processed by standard libraries (i.e., OpenCV [1]) and their audio channel is extracted into Waveform Audio Format files (WAV) in order to be processed by OpenSMILE [3].

As shown in Figure 3, monitoring indications are helping the researcher to always know the state of each task applied to a dataset and to the videos. This aspect is important, as the different tasks (i.e., shot boundaries detection, features extraction, emotion modeling) may take a long time (depending on the size of the processed video) and can lead to issues.



2 - Prepare videos

☑ Overwrite

Videos preparation state (4/6)

Videos

005 Filter by emotion: tenderness

	Name	Emotion	Duration [s]	Monitoring
	005020720	tenderness	2.4	! Preparation ! Shot boundaries detection ✓ Feature extraction ✓ Arousal modeling
	005150814	tenderness	1.72	! Preparation ! Shot boundaries detection

Figure 3: Time consuming tasks are launched asynchronously and their status is updated in real time by push thanks to the WebSockets protocol.

3.2 Features extraction and visualization

Emotions in movies are conveyed through both audio and visual channels. The e*RS* system allows to easily select the desired features to be extracted, depending on the intended objectives. Once extracted, the features can be visualized for each video. As features extraction can be a time consuming task, it is important that the researcher can easily select which features are needed, in order to not extract all the features. The features are then stored in the database, so they can be easily retrieved for a specific video.

Any newly added audio or visual features in the backend will automatically be available for use (selection for features extraction, visualization) in the frontend without any additional line of code.

3.2.1 Audio features

Multiple audio features are available for extraction from the video clips. Those features are inspired from challenges [8, 9] organized in order to create a baseline for affect recognition in audio, as well as from research focusing on emotion recognition in movies [5] and music [2].

The audio features are extracted using the OpenSMILE [3] tool which enables specifying the framing and windowing over an audio file, as well as it gives the possibility to apply functions on the retrieved features. The following list presents the implemented audio features and their descriptions [4]:

[1] https://www.djangoproject.com/
[2] http://www.celeryproject.org/
[3] http://redis.io/
[4] https://angularjs.org/
[5] https://www.websocket.org/

Energy: intensity measuring the distribution of power.

Mel-Frequency Cepstral Coefficients (MFCC): a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. The first 12 coefficients are available.

Pitch (Voice Prob): the voicing probability computed from the ACF(Auto correlation function).

Pitch (F0): the fundamental frequency computed from the Cepstrum.

LSP: the 8 line spectral pair frequencies computed from 8 LPC coefficients.

Loudness and Intensity: the loudness as the normalized intensity raised to a power of 0.3.

Zero-Crossings: the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back.

Spectral (User defined band energies): rectangular summation of FFT magnitudes over the following frequency bands: 0-250, 0-650, 250-650, 1000-4000, 3010-9123.

Spectral Rolloff: the frequency below which X% of the magnitude distribution is concentrated. It is available for the following values of X: 25-50-75-90.

Spectral Flux: the squared difference between the normalized magnitudes of successive spectral distributions.

Spectral Centroid: a measure used in digital signal processing to characterize a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of "brightness" of a sound.

As shown in the literature, functionals describing features and their distribution have a big role in classifying the emotions. The following functions are automatically computed on the features over a window of 1 second: *range*, *skeweness*, *kurtosis*, *minimum*, *maximum* and *mean*.

3.2.2 *Video features*

Two video features are currently supported by the system: shot cut density and brightness. To compute the shot cut density, shot boundaries detection has to be applied first. In order to ease this task, a dedicated view is provided in the web interface. It is possible to configure a list of algorithms (currently supported algorithms are Color Histograms and Edge Change Ratio) and specify a value for the threshold and the weight of the algorithm (if several algorithms are used). Once executed the resulting shot boundaries can be viewed thanks to an interactive visualization linked to the video. It is also possible to provide a ground truth for the shot boundaries values and compute the precision/recall of the resulting shot boundaries detection. As shown in Figure 4, the interactive visualization uses a color box around the playing video, associated to the type of the shot boundary (green for true positives, red for false positives and orange for misses), in order to better understand the result of the algorithm configuration. The shot cut density is then calculated with the following formula as proposed by Wang et al. [10]:

$$c(k) = e^{(1-n(k))/r} \tag{1}$$

where *n(k)* is the number of frames including the *k-th* video frame and r is a constant determining the distribution of the values.

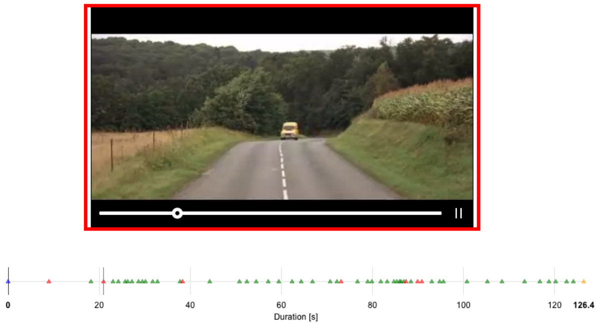


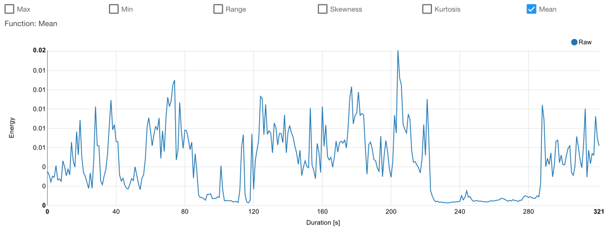
Figure 4: Shot boundaries detection result interactive visualization, with a color box mapped to the type of the shot boundary (true positive, false positive, miss). The video shown here is an excerpt of the movie The Visitors, part of the FilmStim emotional dataset [7].

The brightness is considered as a feature varying with emotions. For instance, dark scenes are used to express negative emotions such as fear and sadness. For each frame, the brightness is calculated over each pixel using the following formula and the mean of these values is computed.

$$brightness(p) = \sqrt{0.299R^2 + 0.587G^2 + 0.144B^2} \tag{2}$$

3.2.3 *Features visualizations*

Plenty of different features can be extracted and for each audio feature several functions are applied, leading to the availability of a huge amount of data. The researcher can choose which features and which functions to visualize (as shown in Figure 5) in order to reduce the amount of data to load. It is therefore possible to focus on the important features and analyze them according to the displayed video.


□ Max □ Min □ Range □ Skewness □ Kurtosis ☑ Mean

Function: Mean

Figure 5: Features functions selection

3.3 Emotion modeling

The emotion modeling is based on an arousal curve generation. Arousal is one one the two main axis of the affective content, with the valence. It corresponds to the intensity of the emotion that is expected to arise in the user while watching the movie.

3.3.1 Arousal curve generation

The arousal is modeled with an approach inspired by the one presented by Wang *et al.* [10]. It consists of a combination of features by applying normalization, smoothing and linear weighted summarization for the fusion. The features can be easily selected among the already extracted features list on the web interface. For instance, the sound energy and the shot cut density are two features to be used for modeling the excitement of movies. Figure 6 shows the arousal curve obtained with these two features for a sequence of Saving Private Ryan (part of the FilmStim dataset [7]).

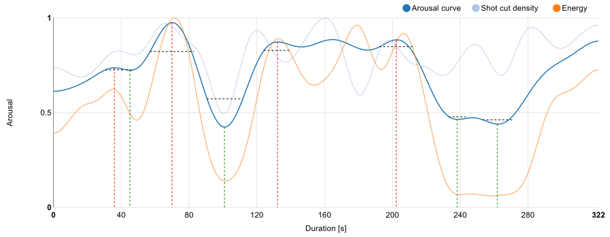


Figure 6: Arousal modeling - arousal curve obtained with sound energy and shot cut density features

3.3.2 High and low arousal partitioning

The arousal curve can be exploited to define more interesting and less interesting parts. To define these scenes we first detect the crests and troughs of the arousal curve. The first derivative of the curve is then calculated in order to determine the transition points around these minima and maxima. Finally the partition between the transition points is restrained so the values contained in it will be near (68%) the minimal or maximal value whether it is a trough or a crest.

3.4 Dynamic visualizations

One important functionality of the e*RS* web application is that all visualizations are linked to the playing video. When the video is played, a timeline is updated on each visible visualization ; it is also possible to click on each of them on a specific position (for instance a peek in the arousal curve) to seek the video at this time. It allows the researcher to analyze the resulting data in the best way.

4. APPLICATIONS

e*RS* has been recently used by students in several Bachelor and Master projects in the University of Applied Sciences of Fribourg, where the goal was to extract emotion information of movies+ and take advantage of this information. For instance, one project used e*RS* to define the less interesting parts of a movie and propose advertisements to the viewer on a second screen application during these moments, without interrupting the display of the movie on the television. Another project maps the arousal level of the movie with the intensity of a dynamic ambient lighting system connected with a smart television, in order to create an immersive movie viewing experience.

We expect our system to be useful to students and researchers who have to manage video datasets for emotion recognition purpose and want to easily test their algorithms and analyze the results.

5. CONCLUSION

We presented e*RS*, an open-source system to facilitate the emotion recognition in movies workflow. The system consists of a backend storing the data and processing the videos, a web application accompanying the researcher through the whole emotion recognition workflow and a REST API making the data available for other applications. In our future work, we will focus on expanding the system by adding valence modeling and classification functionalities and we will also add support for more video features. The detailed information about e*RS* is available on the official project web page[6] and on the GitHub project page[7].

6. ACKNOWLEDGMENTS

We would like to thank Célin Chassot, former bachelor student at HES-SO, for his contribution in the development of some functionalities prototypes.

7. REFERENCES

[1] G. Bradski. *Dr. Dobb's Journal of Software Tools.*

[2] W. C. Chiang, J. S. Wang, and Y. L. Hsu. A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers. *2014 International Symposium on Computer, Consumer and Control*, pages 1249–1252, June 2014.

[3] F. Eyben, F. Weninger, F. Gross, and B. Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 835–838, New York, NY, USA, 2013. ACM.

[4] F. Eyben, M. Woellmer, and B. Schuller. the Munich open Speech and Music Interpretation by Large Space Extraction toolkit. 2010.

[5] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis. A dimensional approach to emotion recognition of speech from movies. pages 65–68, 2009.

[6] A. Hanjalic. Affective video content representation and modeling. *IEEE Transactions on Multimedia*, 7(1):669–676, 2005.

[7] A. Schaefer, F. Nils, X. Sanchez, and P. Philippot. Assessing the effectiveness of a large database of emotion-eliciting films: A new tool for emotion researchers. *Cognition and Emotion*, 24(7):1153–1172, 2010.

[8] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, M. Christian, and S. Narayanan. The INTERSPEECH 2010 Paralinguistic Challenge. In *Proc. Interspeech*, 2010.

[9] M. Valstar, K. Smith, F. Eyben, T. U. München, and R. Cowie. AVEC 2014 - 3D Dimensional Affect and Depression Recognition Challenge. 2014.

[10] Z. Wang, J. Yu, Y. He, and T. Guan. Affection arousal based highlight extraction for soccer video. *Multimedia Tools and Applications*, pages 1–28, July 2013.

[6] http://ers.ilab-research.ch/#/

[7] https://github.com/dumoulinj/ers/