University of Trento

Department of Information Engineering and Computer Science

Multimedia Data Security

September 19, 2017
Academic Year 2016/2017

Discriminate between posed and spontaneous smile

Authors:

Salvatore Manfredi

Emanuele Viglianisi

Tutor

Quoc Tin Phan

Teacher

Giulia Boato

Contents

1. Abstract

Emotion recognition is a very active research area and it is studied in different fields such as linguistic and visual analysis. This paper will present an implementation of a machine learning classifier that discriminates between posed and spontaneous smiles from videos by analysing visual features. The implementation, based on the paper Recognition of Genuine Smiles [1] by Hamdi Dibeklioğlu et al, will be described in different sections of this paper and covers all the steps of the system from the pre-processing to the conclusions.

2. Introduction

In order to start the project it is necessary to give some definitions and keywords. The most important part is defining what a smile is and what we have to focus in order to extract useful features to train the classifier.

A smile is a facial expression, key component of the non-verbal communication, that primarily involves the muscles of the mouth. It can be easily described as "the upward movement of the lip corners" [1, Section 2A].

But the physiological description is far to be simple; according to Ekman and Friesen [2], a smile is the contraction of the zygomatic major, a muscle that extends from each zygomatic arch (cheekbone) to the corners of the mouth.

There are basically two types of smiles: spontaneous and posed. A smile is spontaneous when is driven by an impulse, without premeditation, leading to a smooth movement. Since a smile is the major cue for happiness and enjoyment, people often try to pose it in order to appear more empathetic towards the people around them. The act of posing works by imitating the facial expressions, trying to mimic the contraction of the muscles. Posing, since it is premeditated, leads to a different timing and gradualness of the contractions.

In both posed and spontaneous smiles we can distinguish among three different non-overlapping temporal phases: onset, apex and offset; phases that represent, respectively, the initial, the steady and the final phase of a smile.

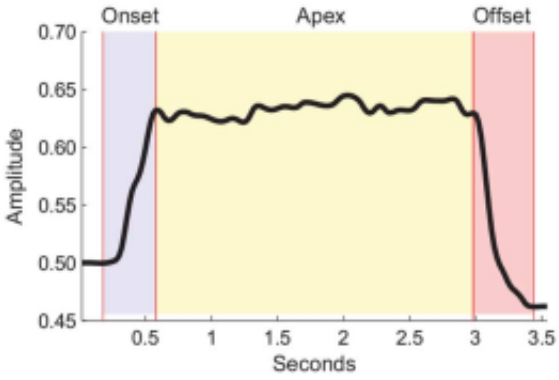


Figure 2.1: Temporal phases of the smile

Dividing in phases is useful because each phase has characteristic movements and features. The onset phase contains the period in which the lip corners distance themselves (following the contraction of the two related muscles). The apex, instead, is a "stability" phase where the distance between the lip corners is steady, while, in the offset, the muscles relax taking the lip corners closer.

Discriminating between posed and spontaneous smiles does not have a practical real-life application yet, although it is not difficult to think a possible integration in the human-computer interaction field. The aim of this study is to understand the effectiveness of recognizing emotions by analyzing facial expression features.

3. State of the art

In the last years, due to the spread of high resolution cameras in everyday devices, many researchers have focused their work on the analysis of emotions from pictures and videos [3][4]. Smile detection and classification is based, like most of the papers in this field, on a machine learning classifier. The most important part, however, is about the extraction of the features from video frames in order to train the learning model.

The researchers who approached the problem of discriminating between posed and genuine smile, used sets of features which differs in both spatial and temporal domains. The paper "How to Distinguish Posed from Spontaneous Smiles using Geometric Features" [5] approach the problem extracting geometric features such as head and shoulders positions, other than tracking the position of landmarks on the face.

The paper "Eyes Do Not Lie: Spontaneous versus Posed Smiles" [6] shows that just tracking the eyelid movements, and other features concerning the eyes, it is possible to train a machine learning model which provides classification rates up to 91 per cent for posed smiles and up to 80 per cent for spontaneous smiles.

A very complete research paper is represented by Recognition of Genuine Smiles [1], Hamdi Dibeklioğlu et al, published in 2015. This paper sums up all the state of the art in features extractions and machine learning classifiers and tests the best methods in a newly created dataset for this purpose [7].

Unlike the aforementioned studies, this paper does not take advantage from geometry features. The system keeps track of the movements of eyes (*Deyelid*), mouth (*Dlip*) and cheeks (*Dcheek*), discarding consequently, with the frontalization process, all those informations related to the head position and orientation.

At this point, the system defines three functions **(x=frame, y=D-values)**, one for each facial region, for example (x=frame, y=Dlip) for what concerns the lip values. Then, it calculates three sets of features (corresponding to the three phases of the smile) for each region. Each set of features contains properties peculiar to the act of smiling; among these cues we can find the maximum amplitude, speed and symmetry. These and more features will be described in the next chapters.

Then, at the end, the sets of features are used to train a machine learning classifier (SVM) and the results of its classification are then compared with the one obtained using different algorithms and datasets from the state of the art. The best result in term of accuracy (0.91) was reached by using all the sets of features merged together (including age and gender) with a 10-fold cross-validation SVM.

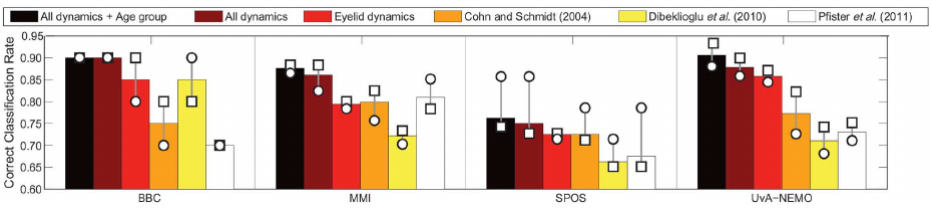


Figure 3.1: Figure from Recognition of Genuine Smiles, performance comparison

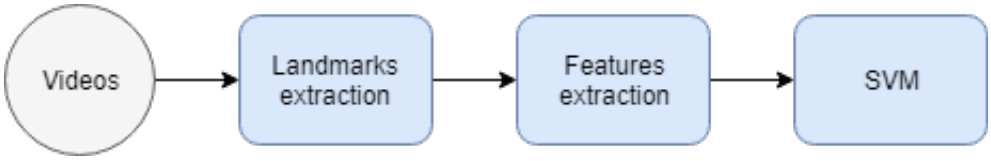
4. Method

Because of its [1] completeness and the numerous recalls to the state of the art, this study aims to implement a method inspired from the one described in the paper Recognition of Genuine Smiles.

The pipeline to create a smile classifier follows these steps

1. expand videos in set of frames;
2. extract 11 face landmarks;
3. frontalize the extracted landmarks;
4. create Dlip and Deyelid functions;
5. split in temporal phases;
6. extract features (25 features per temporal phase) per region;
7. train an SVM classifier and perform a 10-Fold cross-validation;
8. compare the results.

Each of these step will be covered in details in the following subsections.



4.1 Dataset

The dataset used for this project is the UvA-NEMO [7], created by Gevers, Dibeklioglu and Salah as part of the research program named Science Live [8]. The dataset is composed of 1240 smile videos (597 spontaneous and 643 posed), recorded with a resolution of 1920x1080 pixels at 50 FPS with artificial D65 illumination, from 400 subjects with mixed age and sex.

In particular: there are 149 young people and 251 adults, 185 women and 215 men, 43 subjects do not have spontaneous smiles and 32 subjects have no posed smile samples.

Even if the dataset has been created specifically for the purpose of smile discrimination, thus having optimal conditions like: video beginning and ending with a (near-)neutral expression, optimal illumination, high-resolution capture and subjects directly facing the camera, these conditions could also represent an issue. In fact, the model and the algorithms may not work as expected when the system has to process a video with different constraints or not optimal conditions such as everyday scenarios.

Figure 4.1: Figure from Recognition of Genuine Smiles, samples from the UvA-NEMO

4.2 Landmarks extraction

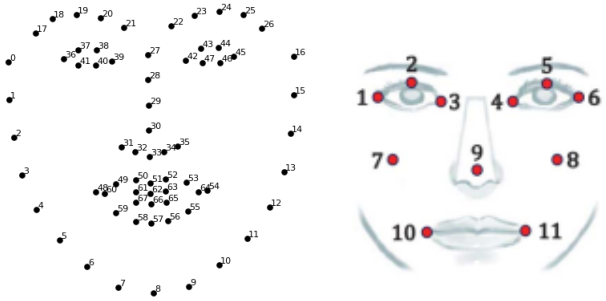


Figure 4.2: Figure (a) shows the 68 detectable points, figure (b) the one we use

The first step of the pipeline is to divide each video into a sequence of frames and then extract the landmarks of the face. In order to perform this action, this study uses the library dlib [9]. Dlib is a set of tools and utilities which allow, using an already trained machine learning model called shape_predictor_68_face_landmarks.dat [10] (trained on the ibug 300-W dataset [11]), to detect 68 face landmarks [Fig (a)]. The landmarks of each frame are saved into a file and will be used in the feature extraction step. However, according to [1], only the 11 points shown in the image above [Fig (b)] are needed.

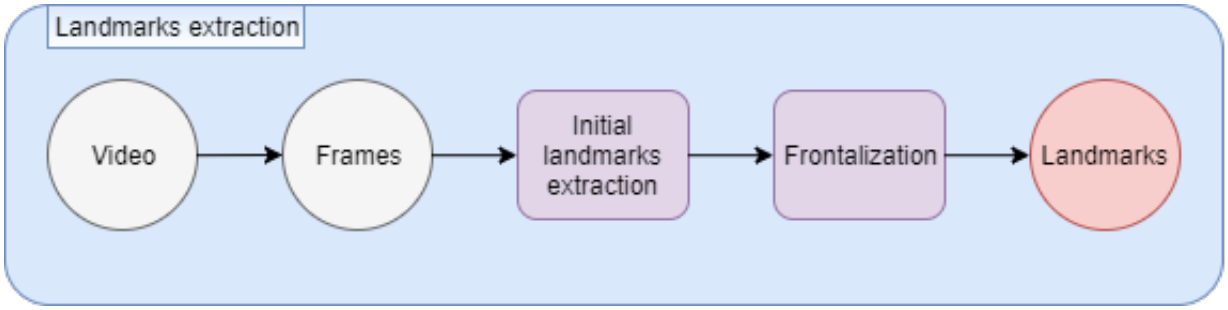


Figure 4.3: Landmark extraction: pipeline

4.3 Frontalization

Even if each video in UvA-NEMO has ideal characteristic of illumination, pose, etc, it is necessary to adapt the algorithm so that it is fit also in a scenario where the subject is not positioned exactly in front of a camera. This because, during the act of smiling, the subject may not stare at the camera for all the time or may turn his head for a brief period of time. These conditions are frequent, especially in spontaneous smiles.

For this reason, the extracted landmarks have to be placed in a fixed 3D mesh face which is then adjusted (through rotation, shifting, etc..) in order to be posed in front of an ideal camera. This process is called frontalization. To perform the frontalization the following methods have been tested:

4.3.1 First Method

The first method uses an implementation [12] of the paper **Effective Face Frontalization in Unconstrained Images** [13] published in 2014.

The system takes a frame as input from which detects and crops the face. The face is then scaled to a standard coordinate system. After this step, the algorithm locates the landmarks and uses them to map the face to a 3D model of a generic face. Quoting from the original paper: "An initial frontalized face is obtained by back-projecting the appearance (colors) of the query photo to the reference coordinate system using the 3D surface as a proxy. A final result is produced by borrowing appearances from corresponding symmetric sides of the face wherever facial features are poorly visible due to the query's pose."

The tests shows that there are many frontalized images with some deformities. This could be caused by the excessive rotation of the head, an incorrect mapping or from the applied algorithm based on the symmetry of the face. Since the interested features to extract are very sensitive to little variations on the frontalized faces, this method proved to be not suitable.

Figure 4.4: Correct and incorrect frontalizations using the first method

4.3.2 Second Method

The system discussed in **Learning Spatially-Smooth Mappings in Non-Rigid Structure from Motion** [14] works in a slightly different way. Expanding the content of the abstract, the method can be described in the following way: given a set of 2D images, instead of applying the face as a "mask" over a single 3D model, the algorithm models each shape (face) in 3D. This mapping is learned in a n-dimensional space of a RIK (rotational invariant kernel) where the smoothness is intrinsically defined; in this way, the model represents the shape variations.

The resulting kernel-based mapping leads to the main advantage of this method: for a newly observed 2D shape, its 3D shape is recovered by evaluating the learned function. This method has undoubtedly better results than the previous one and, among the other things, is exploits the temporal ordering of the variation. In fact, this method considers the face movements as spatial variations in shape space in order to enforce spatial smoothness.

Figure 4.5: Frontalization using the second method

4.3.3 Dlip and Deyelid

The system described here analyzes two of the three areas proposed in the paper [1]. In particular, it focuses on the lips and eyelids features, intentionally ignoring the cheek areas due to the fact that the known algorithm to detect the facial patch cannot be used (due to its license).

4.3.3.1 Dlip

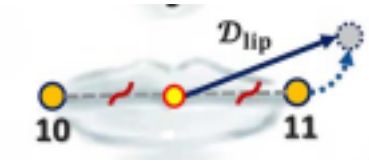


Figure 4.6: Dlip

The Dlip is the average amplitude of the lip corners normalized by the length of the lip. In can be calculated with the following formula:

$$\mathcal{D}_{lip}(t) = \frac{\rho\left(\frac{l^1_{10}+l^1_{11}}{2}, l^t_{10}\right) + \rho\left(\frac{l^1_{10}+l^1_{11}}{2}, l^t_{11}\right)}{2\rho\left(l^1_{10}, l^1_{11}\right)}$$

where "1" indicates the points taken from the first frame of the video, "t" indicates the ones taken from the i-eth frame and ρ is the Euclidean distance.

4.3.3.2 Deyelid

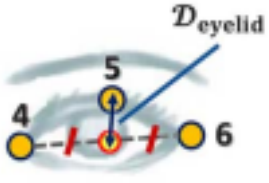


Figure 4.7: Deyelid

As discussed in the second chapter, the contraction of the muscle that closes the lids can be used as a marker to understand whether a smile is posed or not. For this reason the system also calculates *Deyelid*, the normalized eyelid aperture, with the following formula:

$$\mathcal{D}_{eyelid}(t) = \frac{\tau\left(\frac{l_1^t + l_3^t}{2}, l_2^t\right) + \tau\left(\frac{l_4^t + l_6^t}{2}, l_5^t\right)}{2\rho\left(l_1^t, l_3^t\right)}$$

where $\tau\left(l_i, l_j\right) = k\left(l_i, l_j\right)\rho\left(l_i, l_j\right)$ (with k equal to -1 if l_j is located below l_i, 1 otherwise).

4.3.4 Functions and temporal phases division

It is now possible to calculate the functions: **f(frame, Deyelid)** and **f(frame, Dlip)**.

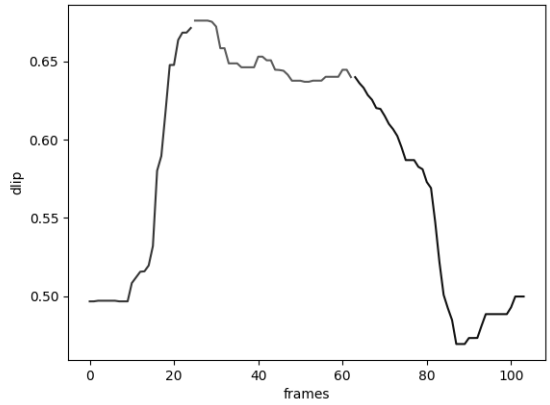


Figure 4.8: Plot of a Dlip function

The image above shows a plot of a Dlip function. It is possible to identify the three temporal phases: onset, offset, apex discussed in the introduction. However, in some cases, the division is not always clearly defined and the discrimination becomes more difficult, especially in spontaneous smiles. For this reason, two main methods have been used: paper-based and clustering-based.

The first method requires to split the function in segments and relies on the assumptions that the onset is the phase that starts from the beginning of the function and finishes at the end of the longest sequence of positive (increasing) segments. The offset, on the other hand, is the phase that goes from the beginning of the longest negative sequence of segments to the end of the function. The apex is the remaining portion of the function. However, this algorithm may fail in different scenarios, for example, if the longest negative sequence ends before the beginning of the longest positive sequence, or if the apex portion is composed only by few points. In order to deal with these kinds of problems, a clustering based algorithm is applied instead. This algorithm exploits the characteristic that spatially near points (in x and y) are likely part of the same temporal phase. The algorithm K-Means is therefore used to creates 3 clusters of points.

The functions could be wrinkled due to the noise in the videos and/or possible fluctuations of the Dlip and Deyelid values. For this reason, these functions are smoothed using a median filter.

4.4 Features Extraction

The part of our system that has the task to extract the features is referred to as the "feature extractor": It takes in input the functions related to Dlip and Deyelid and writes (in csv format) the resulting features, extracted for every temporal phase and facial region.

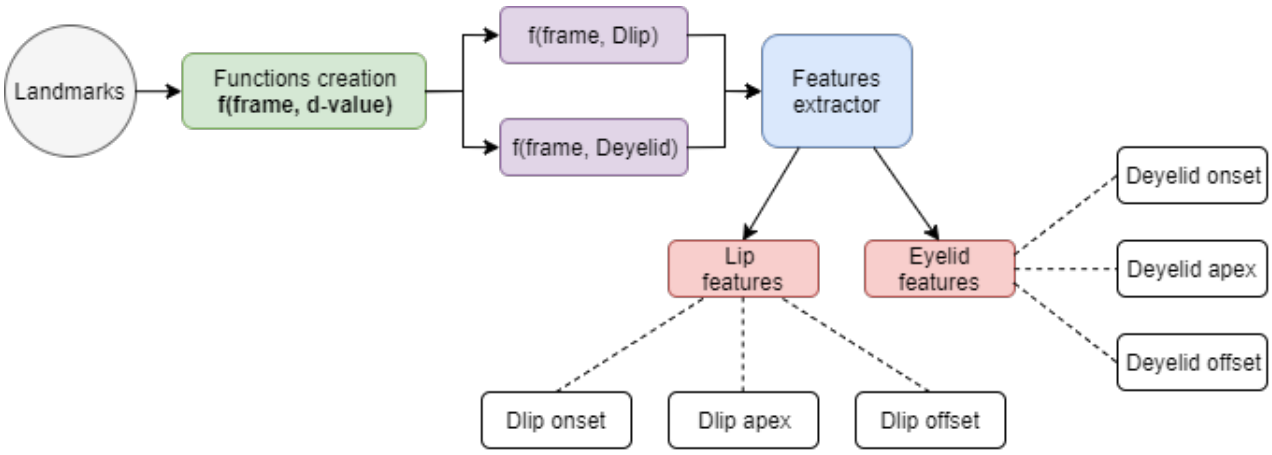


Figure 4.9: Feature extraction: pipeline

Each temporal phase's feature set (related to each facial region) is finally concatenated in order to create a single file (e.g. "Lip features" shown in the image 4.9)

The feature to extract are reported in the table 4.10. The description of the table and any further explanation has been omitted for brevity. Please refer to the [1, section 3B].

Feature	Definition
Duration^{d}:	$\left[\frac{\eta(\mathcal{D}^+)}{\omega} , \frac{\eta(\mathcal{D}^-)}{\omega} , \frac{\eta(\mathcal{D})}{\omega} \right]$
Duration Ratio^{d}:	$\left[\frac{\eta(\mathcal{D}^+)}{\eta(\mathcal{D})} , \frac{\eta(\mathcal{D}^-)}{\eta(\mathcal{D})} \right]$
Maximum Amplitude^{d,m}:	$\max(\mathcal{D})$
Mean Amplitude^{d,m}:	$\left[\frac{\sum \mathcal{D}}{\eta(\mathcal{D})} , \frac{\sum \mathcal{D}^+}{\eta(\mathcal{D}^+)} , \frac{\sum \lvert\mathcal{D}^-\rvert}{\eta(\mathcal{D}^-)} \right]$
STD of Amplitude^{d}:	$\mathrm{std}(\mathcal{D})$
Total Amplitude^{d}:	$\left[\sum \mathcal{D}^+ , \sum \lvert\mathcal{D}^-\rvert \right]$
Net Amplitude^{d}:	$\sum \mathcal{D}^+ - \sum \lvert\mathcal{D}^-\rvert$
Amplitude Ratio^{d}:	$\left[\frac{\sum \mathcal{D}^+}{\sum \mathcal{D}^+ + \sum \lvert\mathcal{D}^-\rvert} , \frac{\sum \lvert\mathcal{D}^-\rvert}{\sum \mathcal{D}^+ + \sum \lvert\mathcal{D}^-\rvert} \right]$
Maximum Speed^{d}:	$\left[\max(\mathcal{V}^+) , \max(\lvert\mathcal{V}^-\rvert) \right]$
Mean Speed^{d}:	$\left[\frac{\sum \mathcal{V}^+}{\eta(\mathcal{V}^+)} , \frac{\sum \lvert\mathcal{V}^-\rvert}{\eta(\mathcal{V}^-)} \right]$
Maximum Acceleration^{d}:	$\left[\max(\mathcal{A}^+) , \max(\lvert\mathcal{A}^-\rvert) \right]$
Mean Acceleration^{d}:	$\left[\frac{\sum \mathcal{A}^+}{\eta(\mathcal{A}^+)} , \frac{\sum \lvert\mathcal{A}^-\rvert}{\eta(\mathcal{A}^-)} \right]$
Net Ampl., Duration Ratio^{d}:	$\frac{(\sum \mathcal{D}^+ - \sum \lvert\mathcal{D}^-\rvert)\omega}{\eta(\mathcal{D})}$
Left/Right Ampl. Difference^{s}:	$\frac{\lvert\sum \mathcal{D}_L - \sum \mathcal{D}_R\rvert}{\eta(\mathcal{D})}$

Figure 4.10: Features to extract

4.5 SVM

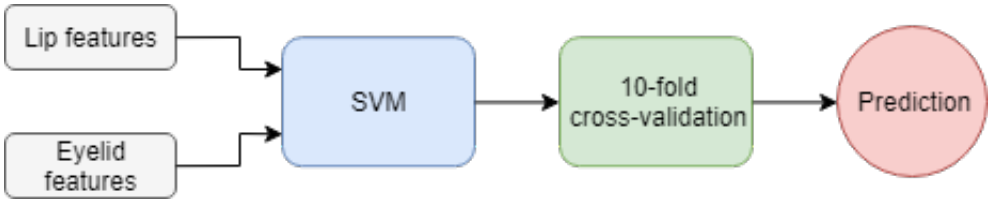


Figure 4.11: SVM: pipeline

The sets of features extracted in the previous chapter are then used to train an SVM classifier. The final accuracy of the system is the output of an outer 10-cross-fold validation which also includes an inner cross validation used to tune the hyper-parameters.

5. Results

The system has performed various tests using different frontalization algorithms, phase division methods and different smoothing "windows" (the number of neighbor points taken into consideration).

In all the tests all the extracted features are used (both eyelid and lip regions with all temporal phases) plus additional information such as gender and age.

Firstly we tried to perform the frontalization using the method introduced in 4.3.1. However, the system performed poorly with an accuracy of 50%. This result could be explained considering that, the applied frontalization process is not optimal due to the aforementioned problems. Some frontalization results contained, indeed, deformations that lead to an incorrect features extraction.

In order to verify that the accuracy was poor due to the wrong results from the frontalization step, the second test was performed using landmarks extracted from non-frontalized frames. The lack of frontalization step was not a big deal because, for the majority of the time, all the subjects in the dataset stare at the camera. The assumption that the issue was caused by the frontalization has been verified with the obtained results. The second test, indeed, obtains an accuracy of 74%, which represents a big improvement compared to the first test.

Given that the first frontalization method is therefore not suitable for these kind of analysis. The second method 4.3.2 for the frontalization was applied obtaining a result of 75%.

During the tests we noticed that a few videos had an incorrect phase separation. In order to overcome this issue, we recomputed the features extraction step for these videos using the cluster-based algorithm for division in phases instead of the implementation of the algorithm mentioned in the paper. Although for those videos we got a better separation, it is difficult to have an algorithm that works at its best in any given case. For this reason, we decided to manually annotate the best division algorithm for each video. Passing over the pair of diagrams generated from the plot of the two algorithm results we noticed that there are cases where both algorithm performed an incorrect separation, especially for the spontaneous smiles, in which there is no clear division of the phases.

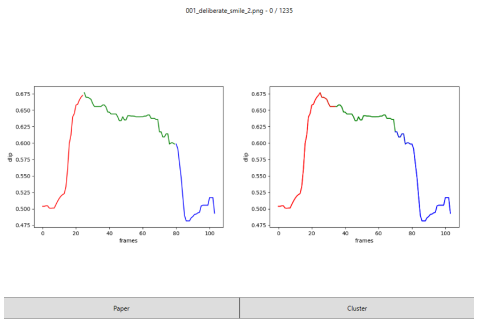


Figure 5.1: Image of the software used for the manual selection

Selecting for each video the best temporal phase division algorithm, using the latter frontalization method and including median filter (with a window of 25), the system's accuracy slightly improved up to **78,3%**, outperforming all the previous test scores.

6. Conclusions

The best result has an accuracy of 78,3% and a F1-score of 80, and it is obtained using the second frontalization method, the best phase division manually selected for each of the videos, median filter with a window of 25, extracting a total of 152 features per video: 25 per temporal phase, both for eye features and lip features, age and gender.

The confusion matrix in 6.1 shows how well the examples have been classified with respect to the relative classes. As can be seen, currently the algorithm misclassifies some spontaneous smiles as posed smiles. This means that some examples had features similar to the one that identify the posed smiles. Probably because, in these examples, the temporal phases were well defined like, as they usually are in the case of posed smiles, and dynamics features like acceleration, mean amplitude etc were pretty much the same.

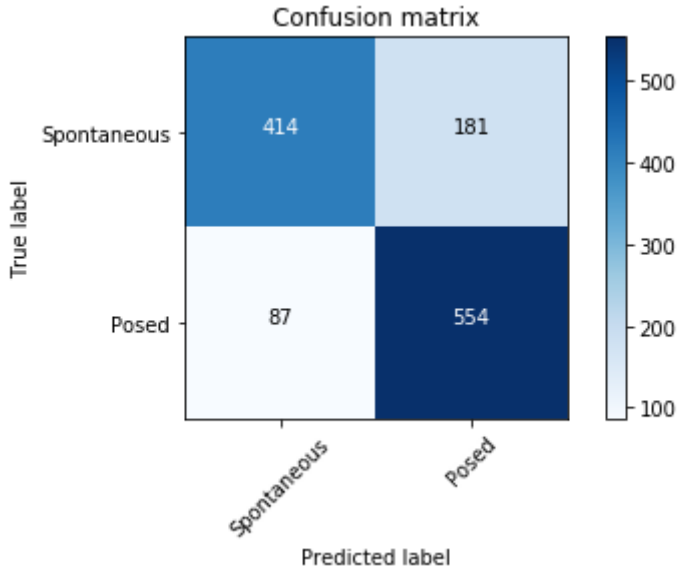


Figure 6.1: Confusion Matrix

The resulting classification is therefore less accurate than the one reported in the paper [1]. This difference could be explained by a not-optimal frontalization process followed by an incorrect temporal phase division. We have evidence of a non optimal phase division when, during the aforementioned manual selection of the division algorithm, we noticed that there were cases in which none of the used algorithms gave an optimal result.

Devising a better phase division algorithm is just one of the possible improvement to the system. In fact, there are many possible changes that can be done and tested. Some of them are:

- test with different feature concatenation technique, passing from the currently used early-level method to a midlevel or a late fusion [1, 5C];
- perform the classification using ranges of age (e.g. 30-40) instead of the specific age;

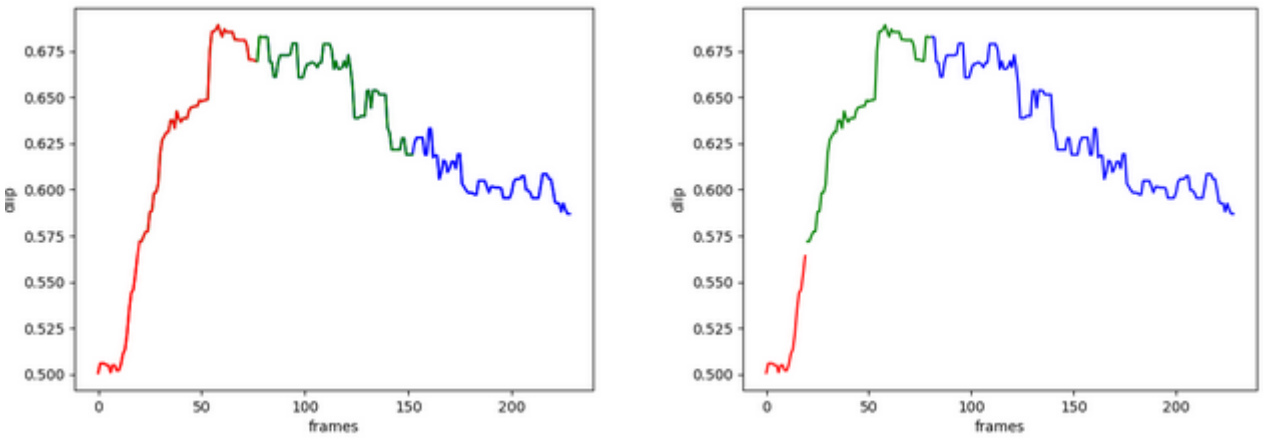


Figure 6.2: The cluster-based method (right) detected the correct sequences

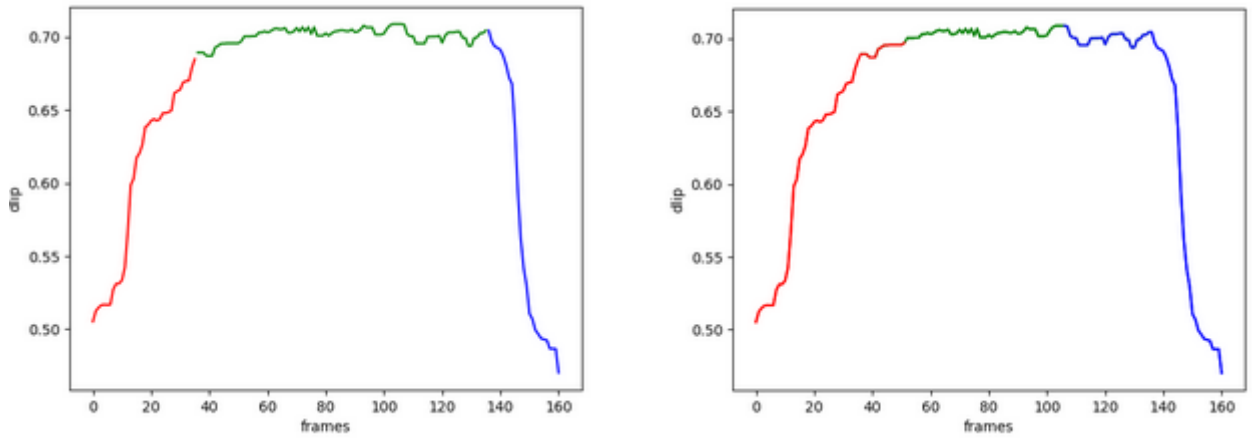


Figure 6.3: The paper-based method (left) detected the correct sequences

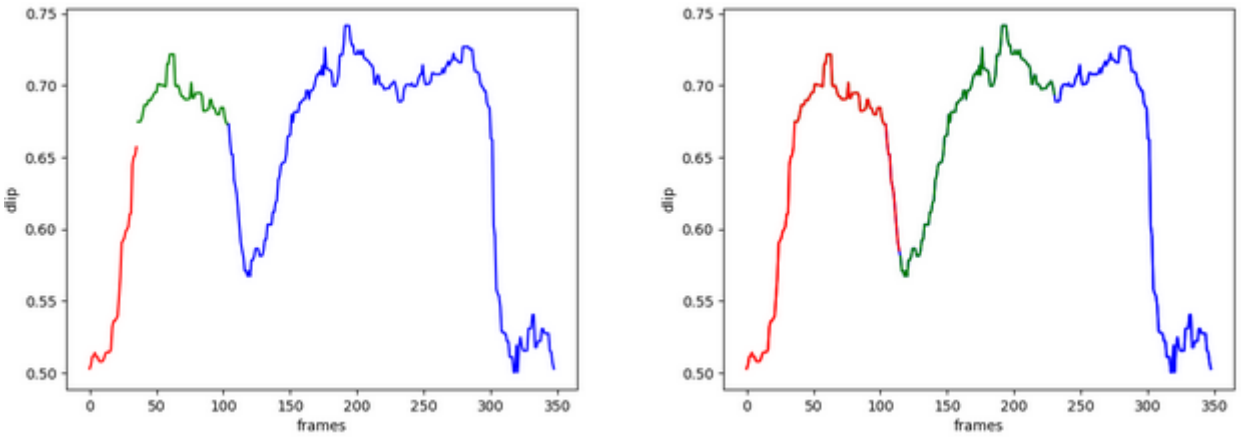


Figure 6.4: Both of the algorithms did not detect the correct sequences

- improve the DLib integration. This can be done creating a new lightweight dlib model using only the 11 needed points (instead of the current 68), reducing the overhead and boosting the performances. This kind of improvement can be also exploited in order to create an end-user application which is able to extract the landmarks and process the features extraction from a video in a short amount of time. This way the results could be quickly evaluated and even be available in real-time.

Bibliography

[1] H. Dibeklioğlu, A. A. Salah, and T. Gevers. Recognition of genuine smiles. *IEEE Transactions on Multimedia*, 17(3):279–294, March 2015.

[2] Paul Ekman and Wallace V. Friesen. Felt, false, and miserable smiles. *Journal of Nonverbal Behavior*, 6(4):238–252, Jun 1982.

[3] D. Dagar, A. Hudait, H. K. Tripathy, and M. N. Das. Automatic emotion detection model from facial expression. In *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pages 77–85, May 2016.

[4] T. Matlovic, P. Gaspar, R. Moro, J. Simko, and M. Bielikova. Emotions detection using facial expressions recognition and eeg. In *2016 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 18–23, Oct 2016.

[5] Michel Valstar, Hatice Gunes, and Maja Pantic. How to distinguish posed from spontaneous smiles using geometric features. pages 38–45, 01 2007.

[6] Hamdi Dibeklioglu, Roberto Valenti, Albert Salah, and T Gevers. Eyes do not lie: Spontaneous versus posed smiles. pages 703–706, 10 2010.

[7] Uvanemo. `http://www.uva-nemo.org`.

[8] Sciencelive. `http://www.sciencelive.nl`.

[9] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[10] Davis E. King. dlib models. `https://github.com/davisking/dlib-models`.

[11] ibug dataset. `https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/`.

[12] Face frontalization: a python implementation. `https://github.com/dougsouza/face-frontalization`.

[13] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[14] Onur C. Hamsici, Paulo F. U. Gotardo, and Aleix M. Martinez. *Learning Spatially-Smooth Mappings in Non-Rigid Structure From Motion*, pages 260–273. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.