



Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Automated detection of microexpressions using a hybrid expert system

Scientific Students' Association Report

Author:

Anna Bodnár
Gábor Révy

Advisor:

Dr. Gábor Hullám
Dániel Hadházi

2021

Contents

Kivonat	i
Abstract	ii
1 Introduction	1
2 Background	3
2.1 FACS	3
2.2 Datasets	5
2.2.1 BAUM-1s	5
2.3 Related works	7
2.3.1 Contempt detection	7
2.3.2 Eyebrow raising and frowning	7
2.3.3 Head movement	8
2.3.4 Lip compression detection	9
2.4 Landmark detector	10
2.4.1 Dlib landmark detector	10
2.4.2 PFLD landmark detector	10
3 Implemented methods	12
3.1 Camera shake detection	12
3.2 Contempt detection	14
3.2.1 Enhancing wrinkle lines	14
3.2.2 Skeletonization	15
3.2.3 Removing mustache	15
3.2.4 Selecting best candidate with Hough transformation	17
3.2.5 Calculating wrinkle score	19
3.3 Nod detection	21
3.3.1 Extracting Euler angles	21
3.3.2 Probability density-based change detection	22

3.3.3	Pattern detection after total variation-based denoising	22
3.3.3.1	Total variation-based denoising	22
3.3.3.2	Pattern detection	24
3.3.3.3	Application	25
3.4	Eyeblink movement detection	27
3.5	Lip compression detection	30
3.5.1	Upper lip thickness estimation	30
3.5.2	Ridge detection-based lip compression detection	31
3.5.2.1	Ridge detection	31
3.5.2.2	Application	35
3.5.3	Chasm detection-based lip compression detection	36
3.5.3.1	Chasm detection	37
3.5.3.2	Application	43
4	Evaluation	45
4.1	Contempt detection	45
4.2	Nod detection	48
4.3	Eyeblink movement	49
4.4	Lip compression	51
5	Discussion	52
	Acknowledgements	53
	Bibliography	54

Kivonat

A mikrokifejezések olyan univerzális arckifejezések, melyek minden embernél ugyanazzal a jelentéssel bírnak. Egy másik jellemző tulajdonságuk, hogy csak néhány pillanatig jelennek meg az arcon. A felismerésükhöz jelenleg szakértői tudásra van szükség, ami gátolja a mikrokifejezések széleskörű alkalmazásának elterjedését, emiatt e feladat automatizálása kívánatos lenne.

Egy ember arckifejezései alapján következtethetünk pillanatnyi érzéseire, illetve értelmezhetjük egy adott esemény által kiváltott reakcióit, mint például reakcióit egy előadásra vagy termékre. Továbbá az arckifejezések egyes jegyeit felhasználhatjuk bizonyos mentális betegségek detektálására.

A tavalyi dolgozatunkban bemutatott hibrid szakértői rendszerünk képes volt az arcon megjelenő néhány alapvető mikrokifejezés felismerésére. Lényeges jellemzője volt, hogy landmark pontok meghatározását leszámítva szakértői algoritmusokat alkalmazott az egyes jegyek detektálásához, mivel nem állt rendelkezésre annotációval ellátott megfelelő adathalmaz, amely egy tanuló algoritmus bemeneteként szolgálhatott volna. Idén folytattuk a megkezdett munkát, a korábbi arckifejezés-felismerő megoldásainkat továbbfejlesztettük, illetve kiegészítettük ajakprés, bólintás és megvetésdetekcióval.

A szemöldökfelhúzás felismerését pontosítottuk, a megjelenő mikrokifejezések detektálására és időbeli lokalizálására idősorelemző algoritmusokat terveztünk és implementáltunk. A vizsgált személy arcára egy általános modellt illesztve meghatároztuk annak pozícióját és orientációját (a Perspective-n-Point probléma megoldásával). Ennek eredménye alapján detektálja megoldásunk a bólintásokat. Egy további fejlesztés részeként ránctektáló eljárást terveztünk és implementáltunk, amelynek segítségével az arcon megjelenő megvetés jeleit érzékeltük. A módszerünket végül kiegészítettük ajakprés-detekcióval is. E gesztus felismerése az ajak vastagságának becslése és az ajakrész szélességének meghatározása alapján egy általunk kialakított idősorelemző algoritmussal valósul meg. Munkánkat valós emberekről készült felvételeken is kiértékeltek. A kiértékeléshez olyan felvételeket kerestünk, ahol spontán, nem megjátszott módon jelennek meg a keresett mikrokifejezések. Ehhez a BAUM-1s adathalmazt használtuk fel, internetes podcastok felvételeivel kiegészítve.

Abstract

Microexpressions are universal facial expressions that have the same meaning for all people. Another characteristic feature is that they only appear on the face for a few moments. Currently, their recognition requires expert knowledge, which hinders the widespread use of microexpressions, thus it would be desirable to automate this task.

A person's facial expressions can be used to infer their momentary feelings or to interpret their reactions to an event, such as their reaction to a presentation or a product. In addition, some features of facial expressions can be used to detect certain mental illnesses.

Our hybrid expert system, presented in our thesis last year, was able to recognize some basic facial microexpressions. A key feature was that, apart from landmark point detection, it used expert algorithms to detect individual features, as there was no appropriate annotated dataset available to serve as input to a learning algorithm. This year, we continued the work we started, improving our previous facial expression recognition solutions by adding lip press, nod and contempt detection.

We have refined the eyebrow raising detection, designed and implemented time series analysis algorithms to detect and temporally localize the microexpressions that appear. A general model of the subject's face was fitted to determine its position and orientation (by solving the Perspective-n-Point problem). Based on this result, our solution detects nods. As part of a further development, we designed and implemented a wrinkle detection procedure to detect contempt. Finally, our method was complemented with lip press detection. The detection of this gesture is based on the estimation of lip thickness and the determination of the lip gap width using a time series analysis algorithm we developed. Our work was evaluated on recordings of real people. For the evaluation, we searched for recordings where the microexpressions appear spontaneously and not due to acting. We used the BAUM-1s dataset, supplemented with recordings of internet podcasts.

Chapter 1

Introduction

Microexpressions are the brief projection of inner emotions onto the face. Detecting them is a very difficult task, even for professionals. A non-expert may be able to recognize a single sign, but taking several signs into account at the same time requires a lot of practice and expertise. Automating this task would help their work, and would also allow a wider use. In our previous report [19] we presented some of the components of a larger hybrid system to detect microexpressions. In this work, extensions to the existing modules are presented and also new components are introduced.

This hybrid system takes advantage of both a deep learning and an expert approach. Deep learning-based image processing became overly popular during the last decade. Expressions can be determined even without expert knowledge with high accuracy [51] [66]. The disadvantage of these systems is the lack of explainability, which can be essential if the system is applied for decision support. Another disadvantage is that training a neural network requires a significant amount of high quality annotated data. Furthermore, there is also no guarantee that the neural network can adapt if we want to classify data that is very different from the ones used for learning, for example, in lightning condition, resolution, and camera motion.

We decomposed the complex expressions into smaller muscular movements and focused on detecting these movements separately. Our hybrid system uses machine learning-based landmark detection to localize key areas of the face. In these key areas, expert algorithms are utilized to detect these muscular activities.

In [19] we developed a gaze detector, an eyebrow raising detector, and a mouth shape estimator. Our gaze detection algorithm included several modules: a pupil localizer and pupil size estimator, a blink detector, and a gaze localizer. The position of the eyes is determined based on the landmark points. To detect the pupil, a gradient-based and an isophote-based algorithm were combined. Furthermore, the blink detection utilized a local Hessian-based blob shape estimation method. In terms of the mouth, the openness and the visible lip size were determined simply based on the landmark points, and the shape of the mouths utilized a Hough transformation-based parabola fitting algorithm.

This year we extended our framework with an improved eyebrow raising detector, nod detection, contempt detection, and lip compression detection. The robustness of the eyebrow raising detection was increased by a total variation regularization-based [54] signal denoising method followed by pattern matching.

We also developed a robust nod detector. First, we estimated the head position and orientation. Then, based on the orientation time series, the movements of the head can

be detected. Here, the same pattern matching algorithm (as in the case of eyebrow detection) was utilized. A challenge in head pose estimation was the occasional camera movements. For that task, we developed a camera shake detector. In the case of intensive camera movements, any detected nod is likely to be a false alarm therefore the detection is discarded.

Our contempt detection solution is based on the strengthening of the nasolabial fold. Contempt is expressed with pulling up one side of the mouth, however, this movement can be really small and quick in case of microexpressions and can be easily mixed up with mouth movements caused by speaking. Therefore instead of concentrating on mouth movements we focused on the strengthening of the nasolabial fold. We enhanced the wrinkle lines using Frangi filter [18]. Although the Frangi filter adequately enhanced the nasolabial fold it also enhanced edges on the face, facial hair, and other wrinkles as well. Therefore we designed modifications in order to eliminate these falsely detected regions. First, the result of the Frangi filter was skeletonized to help the later processing steps. The edges of the face, nose, and mouth was avoided using the landmark points. The edges caused by shadows cast on the face were suppressed by customizing the Frangi filter and the enhanced facial hair was removed with a Gabor filter [21] based method. Lastly, the nasolabial fold was selected from other enhanced wrinkles by fitting a section using probabilistic Hough transformation [46] to each wrinkle, and scoring each wrinkle using the position and orientation of this section.

A lip compression detection was also implemented. Here, the time series of the lip thickness and the lip width was investigated. We implemented a parabolic curve based segmentation algorithm to estimate the thickness of the upper lip. The compression gestures from these time series were detected by a modified ridge pattern detection algorithm (called chasm detection). The motivation of the proposed modification was to increase the sensitivity on the symmetry of the pattern to be recognized.

This thesis is structured as follows. In Chapter 2 we describe the background of our work. This includes the Facial Action Coding System (FACS), which is widely accepted as the basis of emotion detection. We review the existing related works and datasets and present the BAUM [70] dataset we have used to fine tune and evaluate our methods. In Chapter 3 we introduce the implemented algorithms, the main concept of each method, application considerations, and provide examples to demonstrate the viability of the solution. In Chapter 4 we evaluate the results of our methods on videos.

Chapter 2

Background

2.1 FACS

People have universal facial expression for basic emotions. They usually activate the same muscles or muscle groups when expressing these emotions. Ekman et al. proposed the Facial Action Coding System [17] (FACS) to organize these muscular activities. They introduced 58 Action Units (AUs): 12 Upper face AUs and 18 Lower face AUs and 8 AUs for head position, 6 AUs for eye position and 14 miscellaneous AUs. Ekman et al. created a manual for facial action coding [17], it contains a detailed description of each AU and a guide to score the intensity of the AUs.

First, we focused on the detection of AU 1+2. AU 1+2 is responsible for raising the eyebrows and usually causes the appearance of wrinkles on the forehead. Activation of AU 1 and AU 2 simultaneously can mean surprise and fear [16]. When expressing surprise, eyebrows are raised, and eyelids are opened wide open. On the other hand, fear causes the eyebrows not only to raise but to be drawn together too, and the eyelids are also tightened.

Then, we concentrated on AU 53, 54. AU 53 and 54 are responsible for vertical head movement. Activation of AU 54 then 53 causes a nod. Nod means agreement in most of the cultures, and shows that the person on the video clip understands and accepts things.

We also focused on the activation of AU 10. Activation of AU 10 can cause the nasolabial fold to appear or deepen, but not the only muscular activity that can cause the strengthening of the fold. The activation of AU 9, 12 can be also responsible [17]. While activation of AU 10 usually means contempt, AU 9 means disgust and AU 12 means happiness [60]. Because we focused on contempt detection, it was important to distinguish between these emotions. We took advantage of the fact that contempt is the only asymmetric emotion of the three.

Lastly, we detected the activation of AU 24. AU 24 is responsible for compressing the lips, which is a sign of stress.






Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
					
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46
					
Lid Droop	Slit	Eyes Closed	Squint	Blink	Wink
Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
					
Nose Wrinkler	Upper Lip Raiser	Nasolabial Deepener	Lip Corner Puller	Cheek Puffer	Dimpler
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
					
Lip Corner Depressor	Lower Lip Depressor	Chin Raiser	Lip Puckerer	Lip Stretcher	Lip Funneler
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28
					
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure 2.1: Upper and lower face action units [69]

2.2 Datasets

To detect microexpressions, we examined the change in facial features over time. Thus, our algorithms require videos as input. There is a wide variety of datasets created to investigate microexpressions, however, their size, main characteristics, and level of labeling differ significantly. There are datasets containing videos collected from the internet, recorded from TV broadcasts, or shot for a specific purpose. The videos may contain a single person or several people simultaneously. The subject matter of the videos is also varied: reactions to videos, reactions during a specific activity (e.g. poker), or the interaction of the people (e.g. during a meeting or an interview). The dataset may contain short videos with video-level labeling, long videos with timestamped labels, or no labels at all. The video can be recorded from one or from several camera angles (possibly with an overview view), capturing the face in profile or slightly sideways. In the following section datasets related to facial microexpressions are reviewed and the BAUM [70] dataset which was utilized is presented in detail.

MEVIEW [28] (MicroExpression VIdEos in the Wild) contains videos collected mostly from TV interviews and poker games. During poker games the players are under a lot of stress, yet try to hide their momentary feelings, thus the possible detection of their microexpressions is an interesting task. The dataset contains a total of 40 video snippets with an average length of 3s played by 16 individuals. The videos are annotated video-level with action units and emotions by a professional annotator. The SAMM [13] (Spontaneous Actions and Micro-Movements) dataset contains the reactions of 32 participants to 19 videos. The videos are recorded at 200 FPS with a high resolution camera under laboratory conditions. The videos were annotated by 3 certified coders. The frame-level annotations contain both action units and emotions, for a total of 159. SMIC [40] (Spontaneous microexpression Database) contains videos of 20 participants' reactions to 16 video clips. The videos are annotated as positive, negative or surprised by two annotators based on the participants' self-reported emotions. The dataset contains a total of 159 annotations. CASME [67] (Chinese Academy of Sciences microexpression), CASME II [68] and CAS(ME)² [52] datasets were created by the same research group. In all three datasets, participants' reactions to videos were recorded under laboratory conditions. CASME and CASME II contains 195 and 247 microexpression samples respectively, annotated with both actions units and emotions. CAS(ME)² was divided into two parts. The first part consists of 87 long videos that contain micro- and macroexpressions. The second part contains 300 cropped macro- and 57 microexpression samples. MMEW [4] (micro-and-macro expression warehouse) consists of videos in which 300 micro and 900 macroexpressions are annotated. Here, too, the reactions of the participants were recorded. The annotations contain actions units as well as emotions. The AMI (Augmented Multi-party Interaction) corpus [47] consists of 100 hour of recording captured using various devices: cameras, microphones, pens and whiteboard capturing devices. The corpus contains real meetings as well as scenario-driven meetings to evoke a wide range of realistic behaviors. The videos are annotated at several levels: transcripts, dialogue acts, topic segmentations, summaries, emotions, head and hand gestures etc.

2.2.1 BAUM-1s

The BAUM-1 [70] dataset contains short annotated facial video clips from 31 subjects. The video clips are annotated with facial expressions: happiness, anger, sadness, disgust, fear, surprise, boredom, contempt, confusion, neutral, thinking, concentrating, and bothered. The video clips can be grouped into two sets: BAUM-1a contains acted and BAUM-1s

contains spontaneous expressions. In BAUM-1s videos were shown and questions were asked from the subjects and their reactions were recorded, later these reactions were cut up and annotated.

We selected the dataset BAUM-1s for evaluating the contempt, nod, and eyebrow raising detection. We chose this dataset because it contained videos from a high variety of subjects; both men and women of different ages. The annotations were also helpful for the initial evaluations of our methods, but all videos were categorized into strictly one category even if there were multiple expressions present on the video clip and it also lacked the frame level annotation. Therefore we re-annotated a subset of the BAUM-1s with the help of a psychologist. We also annotated a subset of the videos frame level to measure how well our methods locate the facial expressions.

Based on the above, the currently available public datasets are not adequate to train a neural network with sufficient generalization capability. However, some of these datasets may be appropriate enough to achieve acceptable results with a hybrid framework that integrates expert knowledge-based algorithms and machine learning. According to the opinion of our psychologist-expert, the annotation of most datasets was inaccurate, lacking the required detailedness to achieve our aims. Furthermore, in some of these datasets, situational reactions were non-realistic or scenario-specific (e.g., the poker player), whose generality is questionable. We did a significant amount of research to find publicly available datasets which are appropriate for our purposes. None of them were entirely adequate in their original form. Eventually, we have decided to use the BAUM dataset whose annotation was augmented by our expert.

2.3 Related works

There are several solutions, both complex systems to detect microexpressions and specific algorithms to perform the tasks discussed in this thesis. In this section, related works are presented.

2.3.1 Contempt detection

Thekkedath and Sedamkar [59] detected contempt (among the 7 basic emotion) using deep neural networks. They tested their work on the CK+ [44] image sequence dataset which lacked neutral image sequences. The best performing neural network they used was ResNet50 [27] which could perfectly distinguish image sequences of contempt from other emotions.

Sénéchal et al. [57] gave a solution for asymmetric lip movement detection, such as smirk and contempt. They crowd-sourced their own dataset consisting of webcam videos. First, they processed each frame of the videos separately: they computed the Histogram of Oriented Gradients [12] (HOG) features for both the original and flipped frame, then they applied an SVM based classifier. Then used this frame-based score to detect asymmetry events. With their solution they achieved 0.49 precision and 0.69 recall on their own dataset.

Avent et al. [3] constructed neural networks for facial expression detection (interest, happiness, sadness, surprise, anger, fear, contempt, and disgust). They created a three layer neural network for detecting each emotion. They evaluated their model on self collected images with posed expressions and achieved a 81% accuracy in contempt detection.

No one used an expert system to detect the contempt expression and there is no previous nasolabial fold detection method available. On the other hand, there are solutions for general wrinkle detection, mostly for medical or pharmaceutical applications:

Ng et al. [49] used a hybrid hessian filter [48] to highlight wrinkles. Then they applied their proposed hessian line tracking to filter the skeleton of the wrinkles.

Xie et al. [65] used Canny edge detection to filter transient wrinkles, then applied deep first search with extra criteria to separate wrinkles. After that, they identified the structure of these wrinkles and used these structures to create candidate wrinkle regions for their trained an Active Appearance Model [9] (AAM) to refine the wrinkle detection, and lastly they used an SVM based classification to distinguish genuine wrinkles from false wrinkles.

2.3.2 Eyebrow raising and frowning

Liu et al. [42] created a system to improve non-manual grammatical markings used in American Sign Language to signal essential grammatical information. Their setup consists of several layers. An Active Shape Model [10] is used to detect the pose of the head and the face is then warped to frontal view. Using the landmarks from the face tracker geometrical (eyebrow-eye distances) and textural (using Local Binary Patterns and Gabor filter) features are extracted from the face. After feature selection, Conditional Random Fields [36] model is applied to detect eyebrow raise and head shake.

Rauzy and Goujon [53] used the landmark detector of IntraFace [14] to detect landmark points. These landmark points are then corrected with the translation and rotation of the head. The displacements of the landmarks relative to the landmarks of the neutral face

are linearly combined. By analyzing local maxima in wavelet space the raise and frowning of the eyebrows are detected on the obtained time series. Using signal-to-noise ratio the detections can be sorted by likelihood. Their method was evaluated on videos from the Aix MapTask [23] and the Aix-DVD [29] dataset annotated by themselves. The collected video parts are 2h 50m long and contain 431 eyebrow raisings and 142 eyebrow frownings. Based on the signal-to-noise ratio, the detections are divided into 5 sets (from 'A' to 'E'). Considering the 'AB' detections this method achieved a precision of 0.45 and a recall of 0.36 and if we even add the detection set 'C', the precision is 0.31 and the recall is 0.62 on the IntraFace's input.

Khan et al. [34] used the Dlib [35] landmark detector to extract feature points from the face. Several temporal features of the face were extracted. Using the landmark points distances were computed between the obtained points. The magnitude and the angle of wrinkles around the nose and the mouth were extracted using Canny edge detection. The landmark points were refined using optical flow. After applying dimension reduction to the computed distances and wrinkles, the features are fed into probabilistic neural networks and the results are combined using bootstrap aggregation (bagging). Using this combined approach they could reach 0.92 accuracy on the JAFFE dataset [31]. This dataset contains 213 grayscale images of 10 japanese women posing the 6 basic and the neutral facial expression.

Khan proposed a framework [33] for classifying a person's facial expression based on an image. Using a landmark detector important regions were detected, such as the region of eyebrows, eyes, nose and lips. Using edge detection, the obtained landmarks are corrected, and by corner detection, additional feature points are extracted from these regions. Finally, distances of certain feature points are fed into a multilayer perceptron to make the final prediction. This facial emotion recognizer is able to differentiate between the 6 basic and the neutral emotion. The method was tested on the Karolinska Directed Emotional Faces [45] (KDEF) consisting of 4900 images of 35 male and 35 female, showing the 6 basic and the neutral face emotion. Each expression was photographed twice from 5 different angles. Based on the confusion matrix, the classification accuracy is above 0.84 for each class.

2.3.3 Head movement

Chen et al. [7] introduced a nod detection algorithm. To determine the pose of the head they used a 3D Morphable Model thus some 3D input data is required for this method. Based on the retrieved translation and rotation vectors two types of features were extracted for short time windows. Rotation frequency features were computed to characterize the oscillatory nature of head nods in the time frame. Furthermore, the distance to the rotation axis is determined to avoid false detection caused by the movement of the body. These features are then fed into a support vector machine for classification. The algorithm was evaluated on 5-minute segments from the KTH-Idiap dataset [50]. This dataset contains RGB-D videos of one interviewer and three interviewees applying for fund. They annotated the videos themselves for nods. Their best F-score result is 0.72 with a precision of 0.75 and a recall of 0.69. Tan and Rong [58] created a real-time head shaking and nodding detector. In their system, first the face is located by a cascaded classifier [61] learned using the AdaBoost algorithm. Then, the rough region of the eyes is determined based on the face rectangle and a general head model. The exact position is detected by the same algorithm as the face. Based on the exact coordinates of the eyes, head nods and shakes are detected. This is performed by a hidden Markov model. The algorithm was trained and evaluated

on self-collected videos. The training set contained 80 samples with 37 head nods and 43 head shakes and the evaluation set consisted of 110 samples with 49 head nods and 61 head shakes. The results show that the system achieved an accuracy of 82% for nod detection and 89% for shake detection, thus an overall accuracy of 85%. The algorithm of Wei [63] et al. also utilizes hidden Markov models. Their inputs are 3 dimensional originating from a Kinect from which the pose of the head is determined through a deformable model fitting. Based on the obtained Euler angle differences between two consecutive frames, head nods, head shakes and other head movements are recognized by hidden Markov models. Their training and test samples come from a self-collected dataset: 150 samples with 50 head nods, 50 head shakes and 50 other type of head gestures. The method achieved 86% recognition accuracy. Langholz and Brasher [37] used the depth camera of an iPhone X to collect 3 dimensional data. The Euler angles were calculated using the ARKit [2]. They applied data augmentation (shrinking, stretching) and standardization to the initial dataset whose size increased from 482 train and 54 test samples to 172694 train and 18975 test samples. For classification a shallow recurrent neural network was utilized. The best result 91.78% was achieved using GRU cells.

2.3.4 Lip compression detection

Hamm et al. [26] created a hybrid system to automate the FACS-based detection and analyze the facial expressions of patients suffering from neuropsychiatric disorders. To detect the movement of action units, first the Viola-Jones [30] face detector is used to approximate the region of the face. 159 facial landmarks are determined by a trained Active Shape Model (ASM). This large number of landmarks makes it possible to detect fine facial movements. A Kalman filter is utilized to combine the output of the ASM and a temporal model [62]. The movement of the action units is detected by extracting geometric (i.e. displacement of facial parts) and textural (e.g. the appearance of wrinkles) features. To extract textural features, Gabor filters were utilized. Finally, total of 15 Adaboost [20] classifiers were trained to detect 15 action units separately. Their system achieved an average accuracy of 95.9%. The detection accuracy of all the lip-related action units (AUs 10, 12, 15, 18, 20, 23, 25) movements is over 95.7%.

2.4 Landmark detector

Detecting facial landmarks is a crucial step as landmarks serve as reference points from most of our algorithms in the hybrid expert system. In the following sections we describe two alternative implementations, which we investigated and applied.

2.4.1 Dlib landmark detector

One of the landmark detectors utilized can be found in the Dlib machine learning library [35]. It utilizes machine learning-based methods to detect key features on the face. To detect the landmark points, first a face detector is used to detect the exact regions of the faces on the input image. This can also be found in the Dlib library. It utilizes Histogram of Oriented Gradients (HOG) features combined with a linear classifier. Its outputs are rectangles defining regions of faces in the image. The defined regions, along with the image, are fed into the landmark shape predictor. This method determines 68 landmark points on the face as shown in Figure 2.2.

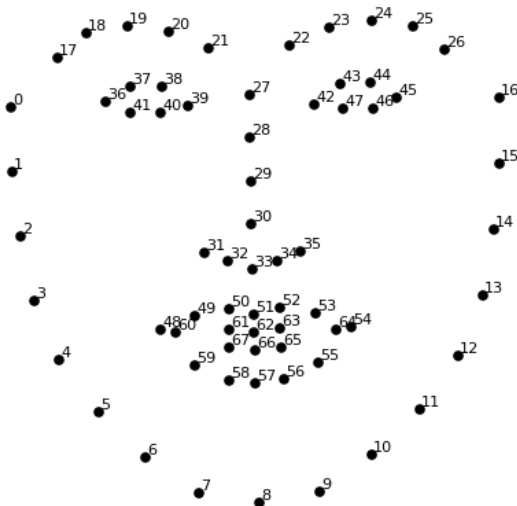


Figure 2.2: The 68 landmark points determined by the Dlib landmark detector.

This predictor uses an algorithm based on an ensemble of regression trees [32]. Its model was trained on the iBUG 300-W [55] facial landmark dataset.

2.4.2 PFLD landmark detector

The Dlib landmark detector operates accurately in the majority of the cases, but it struggled with the detection in some cases. The landmarks were inaccurate or couldn't even be detected, even after correcting the input image (e.g. histogram equalization or contrast increase). After investigating several landmark detector algorithms, PFLD [24] seemed to be accurate enough for our applications. PFLD utilizes a convolutional neural network based on MobileNetV2 [56]. Its input is a cropped image of the face resized to a fixed size on which 106 landmark points are determined. The neural network was trained on the JD-landmark [43] dataset containing 16000 images.

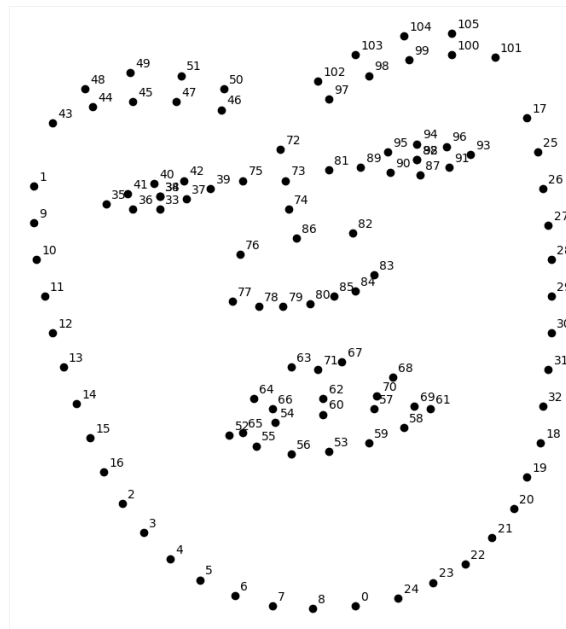


Figure 2.3: The 106 landmark points determined by the PFLD landmark detector.

Chapter 3

Implemented methods

3.1 Camera shake detection

Most of the implemented algorithms are sensitive to the big unintentional movements of the camera. This movement can be detected or compensated for in several ways. For example, the landmark points for the PnP algorithm (described in Section 3.3) were smoothed using first order momentum. This eliminates small detection errors and also the small errors coming from the shaking of the camera. Another approach is to use optical flow to detect the motion of the objects in the background and possibly correct it somehow.

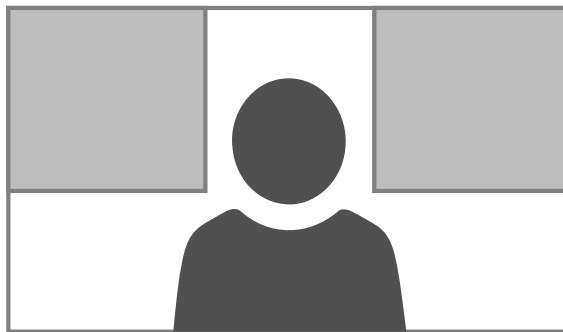


Figure 3.1: The two marked areas are taken into account when detecting camera shake.

In our system a simpler and faster method was utilized. The implemented modules are run on videos that typically feature a person facing the camera. Hence the idea that the shake of the camera can be detected by the movement of the outline of objects in the background. The output of the algorithm can be used either to weight or disable the output of other algorithms or to indicate the fact of the camera shake. The detection is performed by calculating a weighted cross-correlation between the backgrounds of the gradients of successive frames. First, two background areas of the frame are selected based on the detected landmark points: up to the right and left of the head, above the shoulders as shown in Figure 3.1. Next, these areas are cropped to the same size, leaving the appropriate upper corner (i.e. the left upper corner in the left area and the right upper corner in the right area) intact. This brings the cutouts to the same size, the size difference of which is due to the movement of the head. The two areas are then transformed to grayscale. The weighted cross-correlation between two consecutive images on one side

is then calculated as described in the followings. The pixelwise cross-correlation (CC) is computed between two images ($I^{(1)}, I^{(2)}$) as:

$$CC(I^{(1)}, I^{(2)}) = \hat{I}_x^{(1)} \circ \hat{I}_x^{(2)} + \hat{I}_y^{(1)} \circ \hat{I}_y^{(2)},$$

where the \circ operator denotes the Hadamard product. $\hat{I}_d^{(i)}$ denotes the normalized d -direction gradient value of the i th image, where u and v select the row and the column, which can be calculated as:

$$\hat{I}_d^{(i)}(u, v) = \frac{I_d^{(i)}(u, v)}{\|\nabla I^{(i)}(u, v)\|_2},$$

where ∇ denotes the gradient operator, which is calculated using the Sobel operator. u and v select a specific row and column of the image. $\|x\|_2$ denotes the L2 norm:

$$\|\nabla I^{(i)}(u, v)\|_2 = \sqrt{I_x^{(i)2}(u, v) + I_y^{(i)2}(u, v)},$$

Thus the overall weighted cross-correlation is calculated as:

$$WCC(I^{(1)}, I^{(2)}) = \frac{\sum_{S \in \{L, R\}} \sum_{u_S, v_S} CC(I_S^{(1)}, I_S^{(2)}) \circ \|\nabla I_S^{(1)}(u, v)\|_2 \circ \|\nabla I_S^{(2)}(u, v)\|_2}{\sum_{S \in \{L, R\}} \sum_{u_S, v_S} \|\nabla I_S^{(1)}(u, v)\|_2 \circ \|\nabla I_S^{(2)}(u, v)\|_2},$$

where S indicates which side the area is on.

The output of this shake detection is a "similarity" value between 0 and 1 (the higher the value, the more similar the adjacent images). Currently, this value is simply binary thresholded, indicating that the camera was shaken at the given moment.

3.2 Contempt detection

Contempt is one of the seven universal emotions, paired with a universal facial expression of pulling up one corner of mouth [15]. It is the only asymmetric expression among the universal facial expressions. The expression of contempt is achieved by activating the AU 7, AU 10 action units on exactly one side of the face [60]. This often causes the appearance, or strengthening of the nasolabial fold on the corresponding side of the face.

As contempt comes with very small lip movements, instead of detecting the lift of the mouth corner, we concentrated on detecting the strengthening of the nasolabial fold.

3.2.1 Enhancing wrinkle lines

First step of detecting the nasolabial fold was highlighting the wrinkle lines on the face with a customized Frangi filter [18]. Frangi filter is designed to highlight tubular structures on images, such as vessels and wrinkles although it is sensitive to edges as well. Although the original Frangi filter had no problem highlighting the nasolabial fold, it also enhanced several non-wrinkle components: contour of the face, nose and mouth and facial hair if present. To overcome this, we modified the Frangi filter, and applied a Gabor filter based approach as described in Section 3.2.3.

The result of the original Frangi filter for a p pixel of the 2D image is the following:

$$V_p(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp\left(-\frac{R_\beta^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & \end{cases} \quad (3.1)$$

Frangi filter calculates the Hessian matrix with scale s for each pixel, R_β and S are calculated from the eigenvalues of the Hessian matrix ($R_\beta = \frac{\lambda_1}{\lambda_2}$, where $|\lambda_1| \leq |\lambda_2|$, $S = \sqrt{\lambda_1^2 + \lambda_2^2}$). R_β measures the deviation from blob-like structure: R_β is higher for more tubular structure and is lower for blob-like structure. S measures the contrast and is low if there is no structure at p . β and c are parameters of the filter for adjusting the sensitivity of the filter for R_β and S . Authors recommend to apply the filter with a set of s values, and select the maximum intensity result for each pixel:

$$V_p = \max_s V_p(s) \quad (3.2)$$

We modified Equation 3.1 by adding a new term in order to decrease its sensitivity at edges:

$$V_p(S) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp\left(-\frac{R_\beta^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) \frac{1}{1 + \exp(-\gamma(R_\gamma - \alpha))} & \end{cases} \quad (3.3)$$

Where γ and α are parameters, and R_γ can be calculated from the first order derivatives. Let D_x and D_y be the normalized amplitudes of the first order gaussian derivatives with S scale at pixel p . Let also be (I_x, I_y) the normalized eigenvector corresponding to λ_2 also at pixel p and scale S . Then the amplitude of the first order derivative at its maximal direction can be calculated the following way: $D = |D_x I_x + D_y I_y|$. Then R_γ is calculated as $R_\gamma = \left|\frac{\lambda_2}{D}\right|$. The effect of the modification of the Frangi filter can be seen in Figure 3.2.

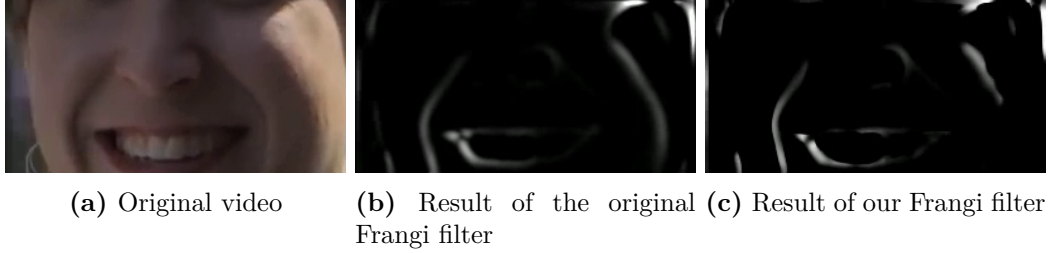


Figure 3.2: The effect of the newly proposed R_γ term. The contour of the shadow is less visible on c) than on b).

3.2.2 Skeletonization

Analyzing the characteristics of the contempt expression, we found that the intensity of the expression depends on the length of the nasolabial fold rather than its thickness. Therefore we applied morphological operations to thin the results of the Frangi filter. The skeletonization also helped at later phases of the processing, as it made it possible to remove unwanted parts of the Frangi image.

During skeletonization we wanted to preserve the central part of the ridges, enhanced by Frangi filter. A pixel p was considered to lay on a ridge if it had greater intensity than its neighbors along the normal direction of the ridge at p :

$$S_p = \begin{cases} 1 & \text{if } V_p \geq (V \oplus K_{I_p}) \wedge V_p \geq T, \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

The maximum intensity of the neighborhood of p was calculated using dilatation. We used four different structuring element (see Figure 3.3). For each pixel we chose the one which direction was orthogonal to the assumed local direction of the ridge (K_{I_p}). The ridge direction was estimated with the normalized the eigenvector $I_p = (I_x, I_y)$ of the larger eigenvalue of the Hessian matrix. To prevent pixels from regions with zero intensity to have non-zero S_p value, we added an extra threshold T for the intensity V_p . This threshold was constant 10 during our experiments. Figure 3.4 shows how the skeletonization works in practice.

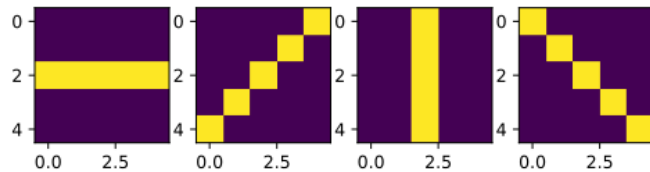


Figure 3.3: Structuring elements used for dilatation.

3.2.3 Removing mustache

As mustache forms ridges on the image, it was usually detected by the Frangi filter. The mustache usually follows the upper lip, but the hair growth direction is mostly vertical. If the mustache is thin, there are might not one but several small ridges following rather the hair growth direction than the upper lip. This case the skeletonized frangi image can

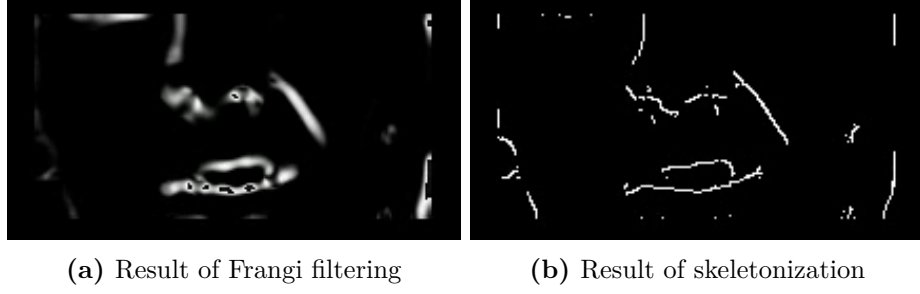


Figure 3.4: Example of skeletonization.

be fragmented at the mustache area (as shown in Figure 3.6a). To remove all fragments of the mustache we connected them with a Gabor filter based approach.

Gabor filter is used to segment features with certain width and orientation on the image [21]. The result of the filter is calculated by convolving the image with the Gabor kernel:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(j\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad (3.5)$$

where $x' = x \cos \theta + y \sin \theta$
 $y' = -x \sin \theta + y \cos \theta$

Where x, y are coordinates. λ is the wavelength of the filter, the higher λ is, the thicker features the Gabor filter highlights. θ is responsible for the orientation of the kernel. ψ is the phase of the kernel, σ is the standard deviation of the Gaussian envelope, and γ is for setting the aspect ratio. And j is the imaginary unit.

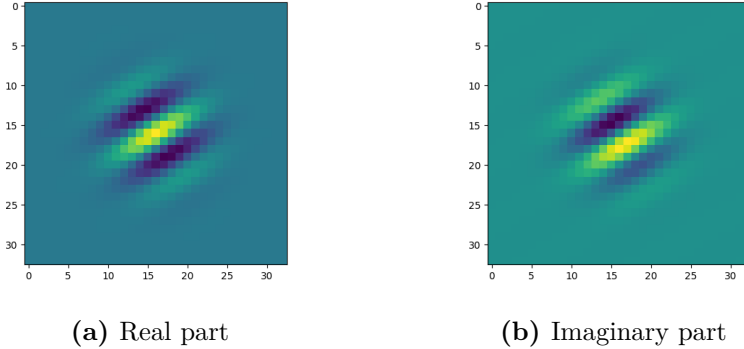


Figure 3.5: Gabor kernel with $\theta = 60^\circ$, $\lambda = 7$ and $\sigma = 3.9669$.

As the mustache can have highly variable form and thickness, we used multiple λ values. σ was set to be $0.5667 \cdot \lambda$ and γ was 1. Furthermore we chose ψ to be zero. That way we enhanced a dark thick line if the result had high absolute value and the phase was below $-\pi + \frac{\pi}{4}$ or above $\pi - \frac{\pi}{4}$. For the mustache region under the nose we used Gabor filter with $\theta = 90^\circ$, and for detecting mustache near the corner of the mouth used $\theta = 90^\circ \pm 15^\circ$. We applied the filter only in the area of interest calculated from the landmarks as shown in Figure 3.6b with blue contour. We wanted to make sure that, this region contains the whole mustache, but small enough for better processing performance. We created it with connecting the 12th, 78th, 84th, 28th, 30th, 61th, 70th, 66th, 52th and 15th landmarks respectively also shown on Figure 3.6b with green circles.



(a) Wrinkles enhanced after skeletonization (b) The areas of interest during mustache removal (c) Results of mustache removal

Figure 3.6: Illustrations for the mustache removal algorithm **a)** shows the wrinkle lines (S) with red on the gray scaled image. **b)** shows the contour of the area in which the Gabor filter was applied in blue, the area between mouth and nose (used in Equation 3.6) with red contour, and the landmarks used for specifying these areas with green circles. **c)** enhanced mustache regions with blue (G), the removed wrinkles with white ($M^* \cap S$) and the preserved wrinkles with red ($W = S - (M^* \cap S)$).

After highlighting the mustache area we had to determine which skeletonized Frangi lines lay in that region. We started from the area between the mouth and nose (Figure 3.6b red contour) and connected the fragmented Frangi lines using the result of the Gabor filter. For that we used morphological operations:

$$M_p^1 = \begin{cases} 1 & p \text{ is between mouth and nose and } S_p = 1, \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

$$M^{i+1} = (M^i \oplus K) \cap (S \cup G) \quad (3.7)$$

Where K is a 5×5 rectangular structuring element and G is the result of the convolution with Gabor filter. We repeated Equation 3.7 until fix point: $M^{i+1} = M^i = M^*$. Then we subtracted the highlighted mustache lines from S :

$$W = S - (M^* \cap S) \quad (3.8)$$

Figure 3.6c shows the mustache removal in practice. With red and white the result of the skeletonization is shown. The output of the Gabor filtering is displayed with blue. The white parts of the skeleton will be emitted in the end of the mustache removal step.

3.2.4 Selecting best candidate with Hough transformation

Elderly people have several wrinkles in the neighborhood of the nasolabial fold. From the several highlighted wrinkle line we wanted to highlight the one most likely to be a nasolabial fold. To identifying individual wrinkle lines on W even if they are fragmented we fitted straight sections on W using probabilistic Hough transformation [46]. We also evaluated for each section how likely it is the nasolabial fold. Let W_h be the point of W lying on a h Hough section.

- the distance of its upper corner form the edge of the nose (s_1),
- its horizontal distance from the edge of the face (s_2),
- its orientation (s_3),

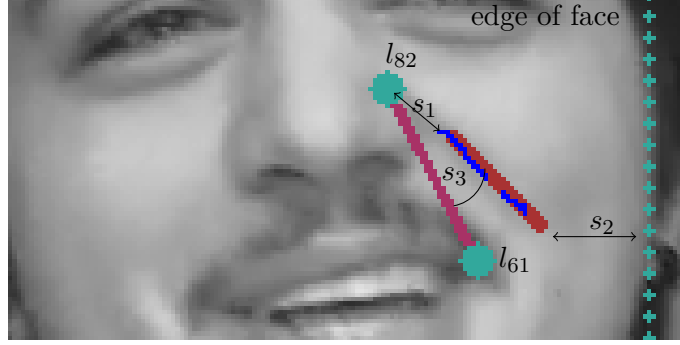


Figure 3.7: Scoring of Hough sections (on the right side): the distance of the upper corner of h for l_{82} (s_1), the distance of the right corner of h from the edge of the face (s_2), the orientation of h relative to line connecting l_{82} and l_{61} (s_3).

- the number of point in W_h (s_4),
- the maximal gap between points of s_h (s_5),

Let h have the endpoints $h_1 = (h_{1x}, h_{1y})$, $h_2 = (h_{2x}, h_{2y})$ where $h_{1y} < h_{2y}$. Let $l_i = (l_{ix}, l_{iy})$ be the i th landmark point detected with the pfd landmark detector. s_1 was determined using the 76th and 82th landmark points (left: $|h_1 - l_{76}|$, right: $|h_1 - l_{82}|$) (see Figure 3.7).

The edge of the face was considered to be the smallest and largest x value of the landmark points. The horizontal distance in s_2 was calculated the following way for the left side: $|\min(h_{1x}, h_{2x}) - \min_i l_{ix}|$ and for the right side: $|\max(h_{1x}, h_{2x}) - \max_i l_{ix}|$ (see Figure 3.7).

h is likely to be the nasolabial fold if it runs from the edge of the nose to the corner of the mouth. We defined s_3 on the left side as the cosine of the angle between the the line fitting on l_{76} and l_{52} and the line fitting to h . For the right side we used l_{82} and l_{61} (see Figure 3.7).

A point p of W was considered to lie on h when its distance from h was smaller than five pixels in that case $\text{proj}_h p$ was added to W_h . Then for s_4 we counted the number of unique points in W_h .

The maximal gap was measured as the maximal distance between neighbor points in W_h . Two points were considered to be neighbors if W_h had no other point on the section between them.

The smaller s_1 , s_3 and s_5 was the more likely h was the nasolabial fold. On the other side, h with larger s_2 and s_4 tended to be the nasolabial fold. The aggregated score sums these scores with weights $\alpha_1, \alpha_2, \dots, \alpha_5$ to select the potential nasolabial fold:

$$h^* = \arg \max_h s(h) = \arg \max_h \sum_{i=1}^5 \alpha_i s_i(h) \quad (3.9)$$

where $\alpha = (-2, 2, 3, 2, -2)$

We only selected h^* if $s(h^*)$ was greater then a threshold to prevent the algorithm from highlighting anything when the nasolabial fold was not present. We also omitted parts of W_{h^*} if it was too close to the edge of the face, mouth or nose or if it was not in the region where we expected the nasolabial fold to appear. This region was defined using the section connecting l_{76} and l_{52} on the left side and l_{82} and l_{61} on the right side. We fitted



Figure 3.8: We expected the nasolabial fold to appear in the square contoured with blue. The red region shows the final search area after omitting the regions near to the edge of the face, mouth or nose.

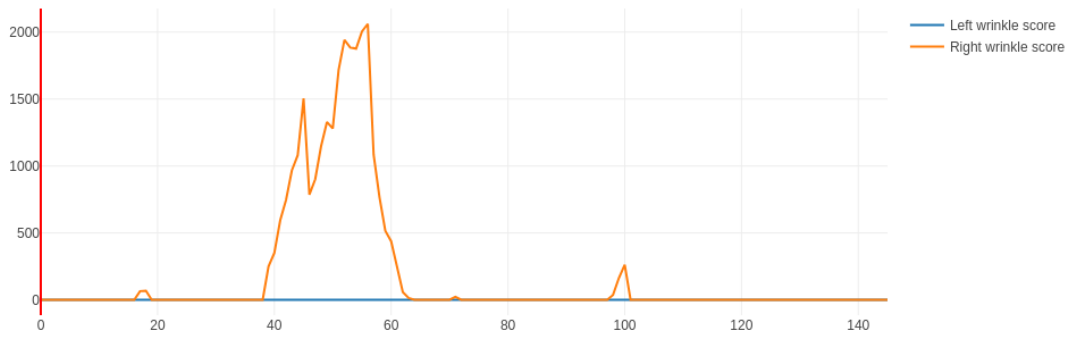
a square on this section as shown with blue contour on Figure 3.8. This square was later reduced with the areas too close to the edge of the face, mouth or nose as shown with red contour on Figure 3.8. We only kept points of W_{h^*} if they were in the reduced square.

3.2.5 Calculating wrinkle score

The last task of the image processing part of the contempt detection was to give a score on the strength of the nasolabial fold on each side of the face. The strength of the nasolabial fold depends on both its length and deepness. The intensity of the Frangi filter result is proportional to the deepness, and the number of pixels in W_{h^*} is proportional to the length. Therefore, we took the weighted sum of points in W_{h^*} on either side of the face:

$$\text{wrinkle score} = \begin{cases} \sum_{p \in W_{h^*}} V_p & \text{if } h^* \text{ was present} \\ 0 & \text{otherwise} \end{cases} \quad (3.10)$$

Figure 3.9 and Figure 3.10 show the contempt detection in practice.

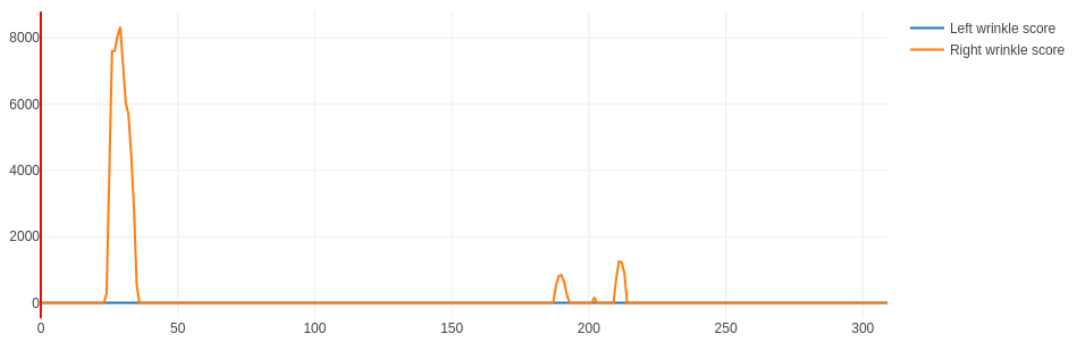


(a) Left and right wrinkle scores.



(b) Frames of the input video at indices 30, 56 and 65. Source: BAUM dataset [70]

Figure 3.9: Example 1 of detecting contempt.



(a) Left and right wrinkle scores.



(b) Frames of the input video at indices 13, 29 and 50. Source: BAUM dataset [70]

Figure 3.10: Example 2 of detecting contempt.

3.3 Nod detection

In this section the investigated and implemented methods for head movement detection are introduced. We focused mostly on detecting head shaking and nodding. The examined methods are evaluated on videos, since the head movement detection requires the analysis of time series patterns of several descriptors. These algorithms are also based on the landmark detector.

3.3.1 Extracting Euler angles

The first step was to determine how certain movements could be robustly recognized. However, based on the displacement of only a few landmark points, the algorithm would not be robust enough. For example, if the person in the video moves their body, it does not imply that they have turned their head. Therefore, the choice was made to examine the Euler angles of the head (the Euler angles are shown in Figure 3.11).

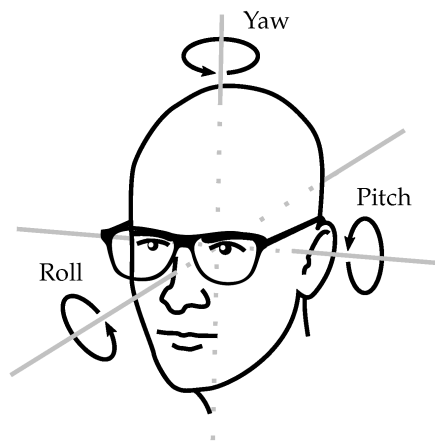


Figure 3.11: Rotational movements of the head expressed in Euler angles. Source: [25]

The problem of determining the pose of a camera based on points in a 2 dimensional projection of the world (i.e. an image) and the corresponding points in a 3 dimensional model of the world is called Perspective-n-Point (PnP) problem. Depending on how we look at the problem, it can also be seen as determining the pose of an object relative to a fixed camera. To calculate the exact pose of an object, at least 4 2D-3D point pairs and a calibrated camera are required, where the 3D points come from an exact model. Usually, a calibrated camera is not available and the 3D model of the head varies from person to person, but with more points we can get a fairly accurate prediction of the pose. There are several algorithms to solve the PnP problem, for example, EPnP, a P3P solver by Gao et al. [22], Infinitesimal Plane-Based Pose Estimation [8] and the RANSAC (RANdom SAMple Consensus) method [38]. After investigating them, an iterative method was chosen, because it had the most stable output. It is implemented in the OpenCV library and utilizes the direct linear transformation algorithm [1] followed by the Levenberg-Marquardt optimization [39]. The inputs of the algorithm are the coordinates of a set of the landmark points and the corresponding 3D coordinates in a simple model of the face based on the adult male anthropometric data¹. The output of the algorithm is the transformation matrix from which the Euler angles and the translation vector can be analytically computed

¹https://en.wikipedia.org/wiki/Human_head (last accessed on 2021.10.25.)

based on the Euler–Rodrigues formula [11]. The resulting time series of the Euler angles is further processed. The examined detection algorithms are presented in the following. During the investigation of the time series we found, that the values oscillate. A significant part of the noise came from the jumps of the landmark points, thus a first order momentum-based smoothing was applied to the coordinates of the landmark points. A first moment vector m_0 is initialized with zeros at time 0. The smoothed landmark points ($\hat{m}(t)$) can be calculated the following way:

$$\begin{aligned} m(0) &= \mathbf{0} \\ m(t) &= \beta \cdot m(t-1) + (1-\beta) \cdot p_{\text{land}}(t) \\ \hat{m}(t) &= \frac{m(t)}{1-\beta^t}, \end{aligned}$$

where t is the time, β is the exponential decay rate for the moment and p_{land} holds the coordinates of the detected landmark points. \hat{m}_t is returned as the smoothed landmark value at time t .

3.3.2 Probability density-based change detection

The first approach we investigated was to use the probability density-based change detection described in the previous scientific students’ association report [19] of our research group. This algorithm detected the changes well, however it had its limitations. People’s heads also move involuntarily a little. This resulted in false detections. This method relies on calculating the parameters of a Gaussian distribution in a sliding window and computing the relative likelihood of the values after this window to the distribution. This technique does not allow for the detection of large changes of the time series’ values in quick succession. This is a result of the sliding window based approach. If the sliding window, based on which the distribution is determined, already contains a significant salience then the corresponding standard deviation may be large. Thus a subsequent change in the signal may be considered insignificant due to a high relative likelihood of values within the sliding window and values after the window, i.e. the likelihood of a significant change will be considered low. This causes problems when detecting consecutive little nods (i.e. the method cannot detect all of them). However, if the size of the sliding window is reduced, even a very little change is considered large. In case of a longer nod, the up and down movements may be detected separately and some patterns could be difficult to recognize. The next approach focused on the pattern recognition.

3.3.3 Pattern detection after total variation-based denoising

3.3.3.1 Total variation-based denoising

Total variation (TV) regularization-based [54] is a noise removal algorithm introduced by Rudin et al. The algorithm aims to minimize the variation of the input series while maintaining fidelity. The variation is defined in our case as the absolute value of the second derivative, while the fidelity is defined as the L1 norm of the difference between the resulting and the original time series. The motivation of the regularization is that we try to smooth the series to become piece-wise linear, which can be achieved by minimizing the L1 norm of the second derivative of the values (based on the theory of Compressed

Sensing [6]). The L1 norm assumes a Laplace likelihood which is robust to outliers (being a heavy-tailed distribution). The second derivative was chosen so that the resulting signal is piece-wise linear. In order to achieve piece-wise linearity, we must minimize the L1 norm of second derivative. The problem can be formulated as:

$$\begin{aligned} \min_{\underline{\mathbf{g}}} \quad & \|\underline{\mathbf{f}} - \underline{\mathbf{g}}\|_1 + \lambda \|\underline{\mathbf{D}} \cdot \underline{\mathbf{g}}\|_1 \\ & \lambda \in \mathbb{R}^+, \quad \underline{\mathbf{D}} : \frac{\partial^2}{\partial x^2} \end{aligned}$$

where $\underline{\mathbf{f}}$ is the input time series, $\underline{\mathbf{g}}$ is the denoised time series, λ is the penalty term of the second derivative and $\underline{\mathbf{D}}$ is the matrix of the discrete second derivative operation.

This can be formulated as a linear program:

$$\begin{aligned} \min \quad & \sum_i e_i^+ + e_i^- + \lambda \cdot \sum_i (d_i^+ + d_i^-) \\ \text{s.t.} \quad & f_i \leq g_i + e_i^+, \quad e_i^+ \geq 0 \quad \forall i \\ & f_i \geq g_i - e_i^-, \quad e_i^- \geq 0 \quad \forall i \\ & g_i'' \leq d_i^+, \quad d_i^+ \geq 0 \quad \forall i \\ & g_i'' \geq -d_i^-, \quad d_i^- \geq 0 \quad \forall i, \end{aligned}$$

where $g_i'' = \frac{g_{i+1} - 2g_i + g_{i-1}}{2}$ is the discrete second derivative value. This optimization problem can be formulated as a Linear Programming (LP) model. The resulting time series is piece-wise linear (see Figure 3.12). Small outliers are also cut; the strength of this effect can be adjusted using the penalty weight of the variation. Using an LP solver to denoise the time series does not scale well, because the size of the input matrices is squared proportional to the size of the input. However, since these matrices are sparse, the ADMM [5] (alternating direction method of multipliers) can be applied. This method was also implemented, and results indicated that it scales adequately. However, details concerning the implementation of this method lie outside the scope of this thesis.

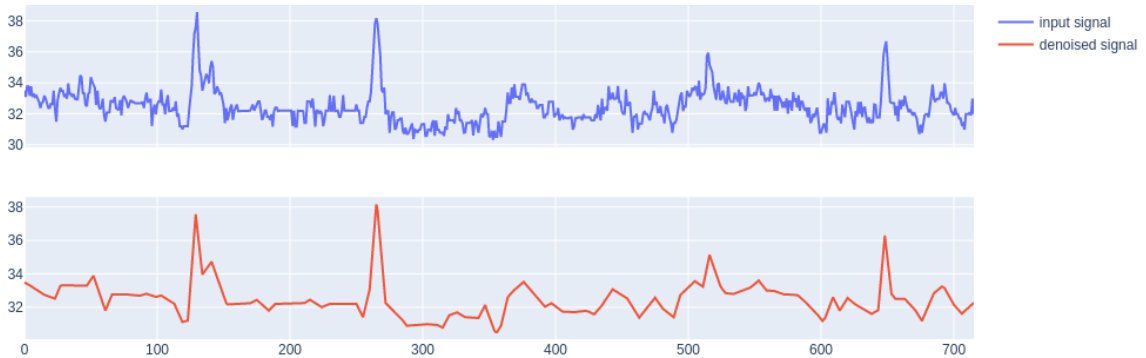


Figure 3.12: Input signal and the result of total variation-denoising. You can see the piece-wise linearity on the result signal.

3.3.3.2 Pattern detection

The developed and implemented pattern detection algorithm is looking for \vee and \wedge shapes of a given maximum width (maximal duration of a single nod) and minimum height (minimal amplitude of a single nod) in the TV-denoised time series. First, the breakpoints between the sections are determined by simply thresholding the second derivative of the time series (resulting in a set of breakpoints \mathbb{B}). In the next step, for each breakpoint (at time t), the nearest breakpoints to the left (t_b) and right (t_a) are determined for which the absolute value of the difference in the time series (y) value is greater than a given threshold ($j > 0$ for jump threshold), and has the same sign (i.e. the deviation is observed in the same direction). To detect the \wedge shapes this can be formulated as:

$$t_b = t - \arg \min_{d: d>0, (t-d) \in \mathbb{B}} y(t) - y(t-d) > j$$

$$t_a = t + \arg \min_{d: d>0, (t+d) \in \mathbb{B}} y(t) - y(t+d) > j$$

This is determined for all the breakpoints except for the first and the last. These \vee and \wedge shaped parts are further filtered by the maximal width to eliminate trends. Since the breakpoints may be far apart from each other, the detected pattern can be considered excessively wide. However, the change of the values within the time series pattern can be well above the threshold j , therefore not all excessively wide detections need to be discarded. First, the detections with a width above a threshold are collected. Then, using linear interpolation between the two outer points (t_b and t_a) and the breakpoints (t_{b+} and t_{a-}) next to the midpoint (t) of the detected pattern, two new outer points (t'_b and t'_a) are determined. These points (t'_b and t'_a) fall between the middle (t) and the previous outer points (left t_b and right t_a respectively) and the amplitude change equals to the jump threshold j .

$$a_1 \cdot y(t_b) + (1 - a_1) \cdot y(t'_{b+}) = y(t) - j \quad \rightarrow$$

$$a_1 = \frac{y(t) - j - y(t'_{b+})}{y(t_b) - y(t'_{b+})}$$

$$t'_b = a_1 \cdot t_b + (1 - a_1) \cdot t_{b+}$$

$$a_2 \cdot y(t_{a-}) + (1 - a_2) \cdot y(t_a) = y(t) - j \quad \rightarrow$$

$$a_2 = \frac{y(t) - j - y(t_a)}{y(t_{a-}) - y(t_a)}$$

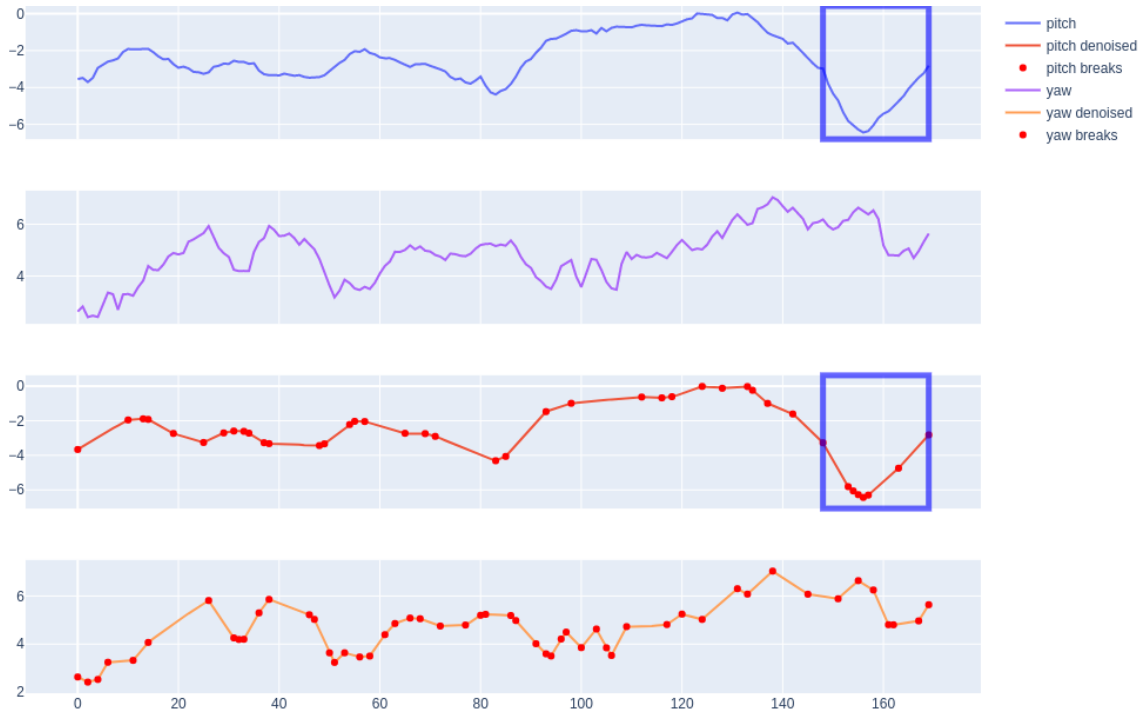
$$t'_a = a_2 \cdot t_{a-} + (1 - a_2) \cdot t_a$$

If the detected pattern based on the new outer points is still considered 'too wide', i.e. higher than a selected threshold, then the detection is discarded, else it is retained.

3.3.3.3 Application

In recognizing the pattern of nodding, it was assumed that there was a back and forth jump on the denoised 'pitch' time series during the nod. This means detecting \vee and \wedge shapes on the time series. After denoising the 'pitch' time series with the TV denoising algorithm, the pattern detection was performed on it.

Examining the results, a false detection was observable when the person in the video accidentally shook the camera. Thus these results are further filtered using the output of the implemented camera shake detector (described in Section 3.1). False detections were also discernible when the person in the video looked sideways-down and then back. These detections are also filtered out using the output of the same pattern recognition algorithm performed on the 'yaw' time series. Figure 3.13 and Figure 3.14 show different scenarios. The plots visualize the original and the denoised pitch and yaw angle time series. The first two rows show the initial detections, and line 3 shows the final nod detection. Figure 3.13 shows a simple scenario: a single nod at the end of the video. There is no 'yaw' movement. Figure 3.14 demonstrates a more complex scenario. There are detections at 121 – 151 and 364 – 393, which are nods. There are also detections between 254 and 299, however they are filtered out since there are also 'yaw' movements at the same time.

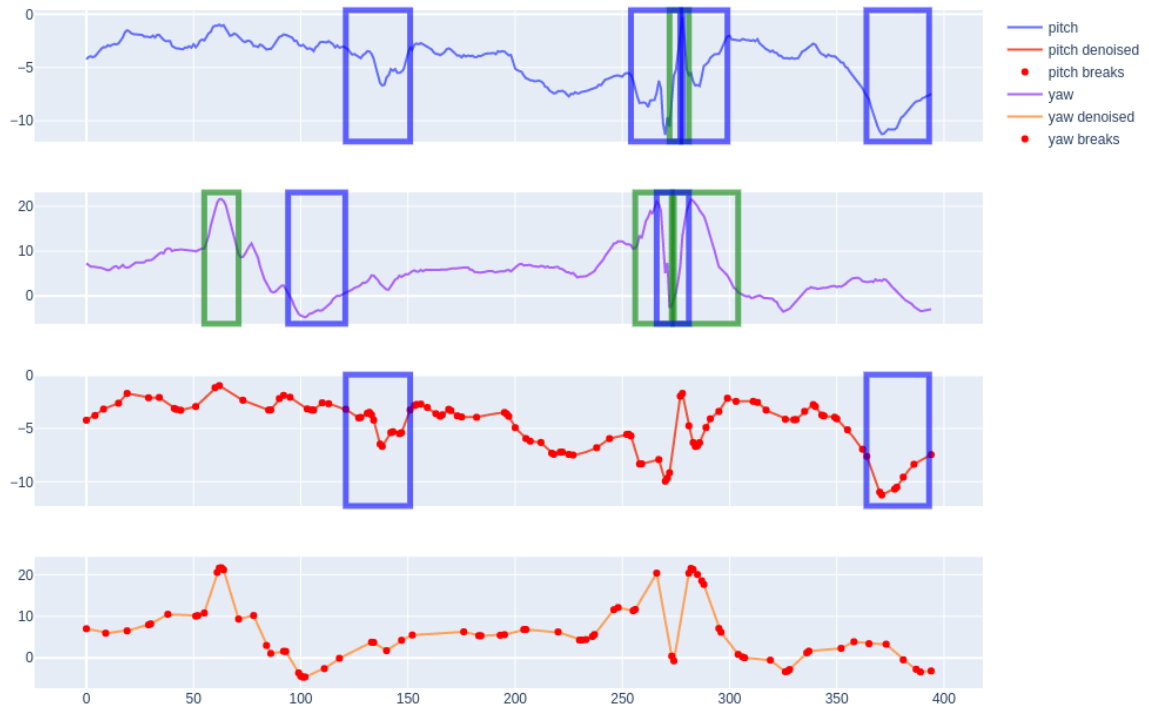


(a) Pitch and yaw angle input signals and the corresponding denoised signals. Red markers mark the breakpoints of the denoised signal and a blue rectangle denotes the detection.



(b) Frames of the input video at indices 148, 156 and 169. Source: BAUM dataset [70]

Figure 3.13: Example 1 of detecting a nod.



(a) Pitch and yaw angle input signals and the corresponding denoised signals. Red markers mark the breakpoints of the denoised signal. Blue (∇) and green (\wedge) rectangles denote detections.



(b) Frames of the input video corresponding to relevant indices of the time series. Source: BAUM dataset [70]

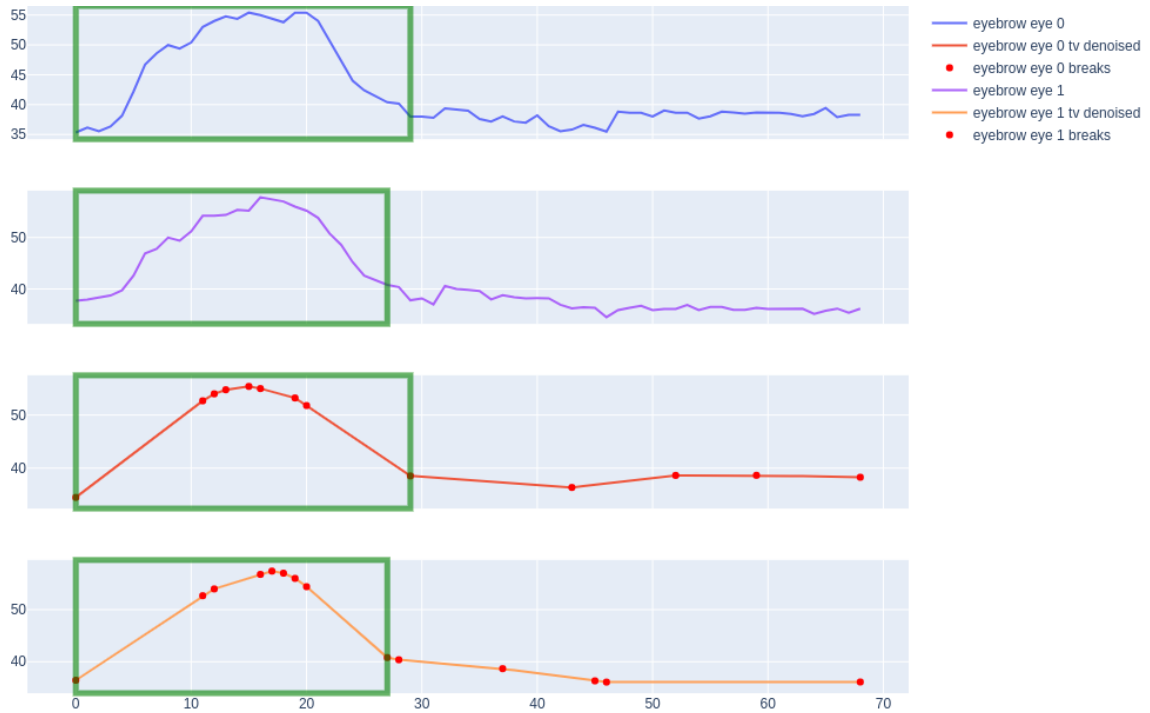
Figure 3.14: Example 2 of detecting a nod.

3.4 Eyebrow movement detection

In this section the detection of eyebrow movements i.e. eyebrow raising and frowning are described. In our previous report [19] two algorithms were introduced to this task: a median filter-based and a probability density-based. Those algorithms performed well on videos recorded under controlled conditions. However, we would like to use our hybrid expert system in real life applications: for videos recorded with web cameras, laptop cameras or the front camera of smartphones. This means, that people may accidentally push, move or shake the camera. Applying the probability density-based detection, the results were not encouraging. In this paper a more stable approach is presented: pattern matching after total variation denoising, which was also utilized to detect head movements (as described in Section 3.3.3). This method also requires videos as input as the change in the eyebrow position is detected. The landmark points are still the main input to the algorithm. However, we found, that the landmark points around the eyes and the eyebrows are more accurate when detecting with the PFLD landmark detector (described in Section 2.4.2). Therefore, the output of the PFLD landmark detector was utilized. This detector also utilizes a camera shake detection algorithm which was introduced to the system as an individual component (see Section 3.1).

The time series on which the eyebrow movements are detected is produced the same way as described in our previous report[19]: at each frame in the video the distances between specified points are recorded and further processed as a time series. The selected points were the middle points of the eyebrows (49 and 104 in Figure 2.3) and the mean of the lower points of both eyes (33, 35, 36, 37, 39 and 87, 89, 90, 91, 93 in Figure 2.3). The bottom points were selected because their position is changed to a lesser extent when blinking, compared to the upper points which are pulled down. The total variation regularization is applied to the resulting time series.

The output of the total variation regularization is piece-wise linear. It is expected, that the short outlier sections are cut by the algorithm, while the main trends are preserved. On such a denoised time series a \wedge (or a \vee) shape can be found during the raising of the eyebrow (or frowning). Thus, such shapes are detected on the time series. The detected shapes are constrained by the minimal and maximal width and the minimal height. The minimal and maximal width is predefined, the minimal height is determined to be 10% of the moving-average of the time series. Although the method measures the distance between the landmark points and not the absolute position, the movement of the camera may cause outliers in the time series due to the perspective change and the inaccuracy of the detection on the blurred image. The false detections caused by the shake of the camera are filtered out using the camera shake detector (described in Section 3.1). Figure 3.15 and Figure 3.16 show examples of the detection. Figure 3.15a shows the time series in which the movement is detected. Figure 3.15b shows the relevant frames from the input video. Figure 3.16a shows an example of two consecutive eyebrow raisings.

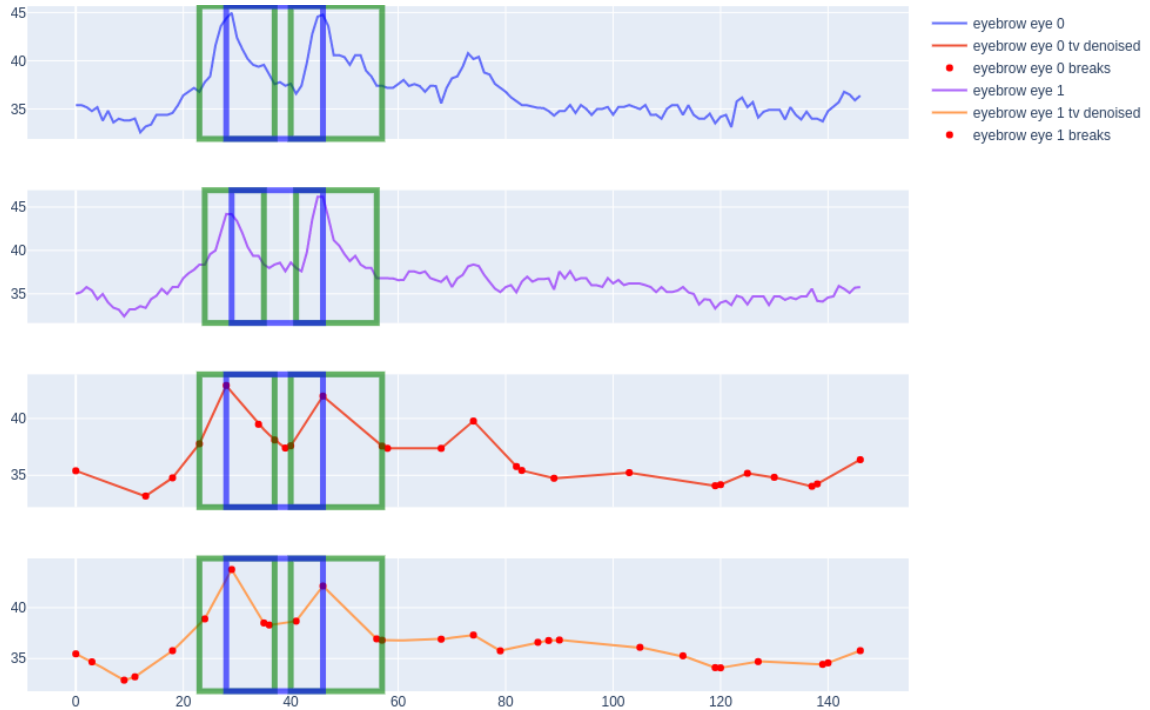


(a) Eyebrow-eye distance (in pixels) input time series and the corresponding denoised time series. Red markers mark the breakpoints of the denoised time series and a green rectangle denotes the detection.



(b) Frames of the input video at indices 0, 15 and 29. Source: BAUM dataset [70]

Figure 3.15: Example 1 of detecting eyebrow raising.



(a) Eyebrow-eye distance (in pixels) input time series and the corresponding denoised time series. Red markers mark the breakpoints of the denoised time series. Blue (∇) and green (\wedge) rectangles denote detections.



(b) Frames of the input video corresponding to relevant indices of the time series. Source: BAUM dataset [70]

Figure 3.16: Example 2 of detecting eyebrow raising.

3.5 Lip compression detection

In this section the implemented methods for lip compression detection are described. The algorithm detects the thinning of the upper lip based on the lip thickness time series. The total variation regularization-based detection (described in Section 3.3.3) could not be applied here, due to the difference between the properties of the patterns to be detected. In case of the nod detection, only the jump in the angle values is important. However, in case of the lip compression detection, the observable pattern is symmetric or nearly symmetric. The estimation process of the lip thickness is described in Section 3.5.1. The first approach implemented for the detection was based on the ridge detector described in Section 3.5.2.1 while the final version utilizes its modified version described in Section 3.5.3.1.

3.5.1 Upper lip thickness estimation

Both the PFLD and the Dlib landmark detectors mark the edges of the lips with many landmark points, however the localization error of these points is usually large. The reason for this is that the training dataset of these detectors contains frontal face photos with a closed mouth. Thus, we try to detect an “outlier” gesture, which cannot be done directly from the landmarks, since they are usually inaccurate in these situations. We designed an algorithm in order to increase the accuracy of this measurement. The inputs of this algorithm are the grayscale image (\mathbf{I}), the coordinates of the inner corner of the lips (marked by \mathbf{c}_{left} and \mathbf{c}_{right}), the coordinates of the midpoint of the upper edge of the upper lip (marked by \mathbf{c}_{up}) and the central line of the lip gap (with normal vector \mathbf{g} and central point \mathbf{c}_{cent}), which is determined by the algorithm detailed in [19]. First, the coordinates and the image are rotated so that the center line of the lip gap becomes horizontal.

$$\begin{aligned}\alpha &= \frac{\pi}{2} - \tan^{-1}\left(\frac{\mathbf{g}_y}{\mathbf{g}_x}\right) \\ \mathbf{c}'_{left} &= \mathbf{R}(\alpha) \cdot \mathbf{c}_{left} \\ \mathbf{c}'_{right} &= \mathbf{R}(\alpha) \cdot \mathbf{c}_{right} \\ \mathbf{c}'_{cent} &= \mathbf{R}(\alpha) \cdot \mathbf{c}_{cent}\end{aligned}$$

Let $\mathbf{R}(\alpha)$ be the rotation matrix with α angle, and \mathbf{I}' be the counterclockwise rotation of \mathbf{I} with α . The thickness is estimated from the segmentation of the upper lip, which is realized by choosing the maximal score of the potential paraboloid segmentation curves. These curves are defined as follows:

$$\begin{aligned}s^{(a,b)}(x) &= \\ &= \begin{cases} \left(x - \mathbf{c}'_{left}(x)\right)\left(x - 2k(a) + \mathbf{c}'_{left}(x)\right)\left(\mathbf{c}'_{up}(y) - \mathbf{c}'_{cent}(y)\right)b & \text{if } \mathbf{c}'_{left}(x) \leq x \leq k(a) \\ \left(x - \mathbf{c}'_{right}(x)\right)\left(x - 2k(a) + \mathbf{c}'_{right}(x)\right)\left(\mathbf{c}'_{up}(y) - \mathbf{c}'_{cent}(y)\right)b & \text{if } \mathbf{c}'_{right}(x) \geq x \geq k(a), \end{cases}\end{aligned}$$

where $k(a) = a \cdot \mathbf{c}'_{left}(x) + (1 - a) \cdot \mathbf{c}'_{right}(x)$ is the x coordinate of the projection of the lip gap in the examined image (let $\mathbf{k}(a) = [k(a); \mathbf{c}'_{cent}(y)]$ be the 2d coordinates of this

point). The motivation of using $k(a)$ instead of the middle of the visible gap line is that the projection of the middle of this gap moves toward to the corner of the lip, if the head is turned to the side.

In order to score the segmentation curves, the projection of the gradient vectors to the radius of the curves are examined:

$$\text{score}(s^{(a,b)}) = \int_{s^{(a,b)}} \nabla \mathbf{I}'(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{k}(a)) d\mathbf{x}$$

The estimated thickness of the upper lip is defined by:

$$-(\mathbf{c}'_{up}(y) - \mathbf{c}'_{cent}(y)) \cdot \mathbf{b}^*,$$

where $(a^*, b^*) = \arg \max_{a \in A, b \in B} \{\text{score}(s^{(a,b)})\}$. The set of the examined values of a is $A = \{0.35, 0.5, 0.65\}$, while for b is $B = \{0.2, 0.5, 0.7, 1.0, 1.2, 1.4\}$.

3.5.2 Ridge detection-based lip compression detection

3.5.2.1 Ridge detection

Ridges are \cup and \cap shapes in a time series. Having an idea of the width of the ridges specific to the application, they can be detected with this algorithm. The idea behind the detection is that by taking the second derivative of the function after proper smoothing, we can see a local extremum with a higher absolute value at the center point of the ridge. The proper smoothing means convolving the function with a Gaussian kernel of adequate σ . To make the kernel scale-independent i.e. to make the result of convolving a ridge with a kernel of appropriate σ independent of its width, the Gaussian kernels are normalized based on the scale space theory [41]. Since derivation is a shift invariant, linear operation, it can be calculated by convolution. It is associative and commutative operation thus normalized second derivative of Gaussian kernels are utilized. First, the input signal is convolved with a filter bank of Gaussian kernels of different σ s.

$$\begin{aligned} C_\sigma(x) &= (y * \widehat{G''_\sigma})(x), \\ \text{where } \widehat{G''_\sigma} &= \frac{\partial^2 G_\sigma}{\partial x^2} \cdot \sigma^2 \\ G_\sigma(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \end{aligned}$$

Here, C_σ holds the result of the convolution of the input signal with a kernel of a particular σ . $\widehat{G''_\sigma}$ is the normalized second derivative of Gaussian kernel ($G_\sigma(x)$), where the the normalization factor σ^2 can be derived based on the scale space theory [41]. In the next step at a given point x the scale with the strongest response can calculated, that is the σ , where $C_\sigma(x)$ is minimal if we are looking for \cap shapes or maximal in case of \cup shapes):

$$\begin{aligned}\Sigma_{min}(x) &= \arg \min_{\sigma} C_{\sigma}(x) \\ \Sigma_{max}(x) &= \arg \max_{\sigma} C_{\sigma}(x)\end{aligned}$$

The significance of this step can be illustrated with the following two figures: Figure 3.17 and Figure 3.18. Figure 3.17a shows an angular ridge, with a width of 12. Red dashed lines mark the important distances (3, 6, and 10) from the center of the ridge. Figure 3.17b contains 3 normalized second derivative of Gaussian filters (with σ values of 3, 6 and 10). The result of the convolution with the 3 kernels is shown in Figure 3.17c. As can be seen, the function created by convolution with the 6- σ kernel has the highest value at the center of the ridge. It can be calculated that the normalized Gaussian kernel with σ parameter intersects the x-axis right at sigma, thus "highlighting" ridges with a radius of σ (based on half width at half maximum estimation):

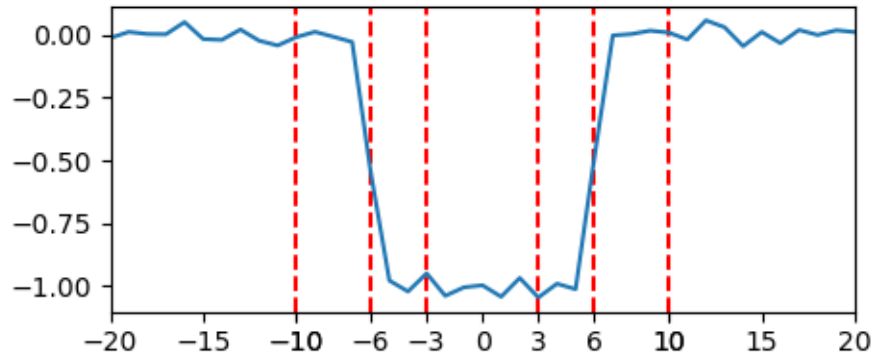
$$\begin{aligned}\frac{\partial^2 G_{\sigma}}{\partial x^2} &= 0 \\ \frac{\partial^2}{\partial x^2} \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) &= 0 \\ \frac{1}{\sqrt{2\pi}\sigma^3} \left(\frac{x^2}{\sigma^2} - 1 \right) e^{-\frac{x^2}{2\sigma^2}} &= 0 \\ x &= \pm\sigma\end{aligned}$$

Figure 3.18 illustrates this detection in another situation. 3.18a visualizes a signal with a rounded ridge with a half width at half maximum of 6, marked with the distances corresponding to the σ values of the kernels. 3.18b visualizes the normalized second derivative of Gaussian kernel with the same σ parameters, and 3.18c shows the result of the convolutions. Again, the convolution value on the signal generated with the 6- σ kernel has the highest absolute value in the center point of the ridge.

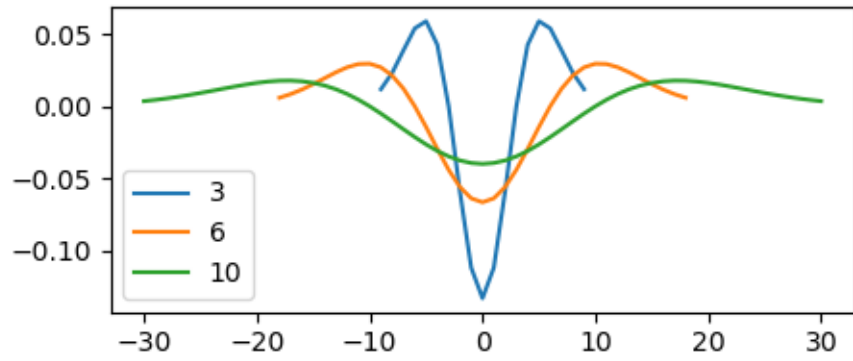
Selecting the appropriate σ in each point, a new signal can be created: the element-wise maximum or minimum of the signals as a result of convolution:

$$\begin{aligned}C_{max}(x) &= \max_{\sigma} C_{\sigma}(x) = C_{\Sigma_{max}(x)}(x) \\ C_{min}(x) &= \min_{\sigma} C_{\sigma}(x) = C_{\Sigma_{min}(x)}(x)\end{aligned}$$

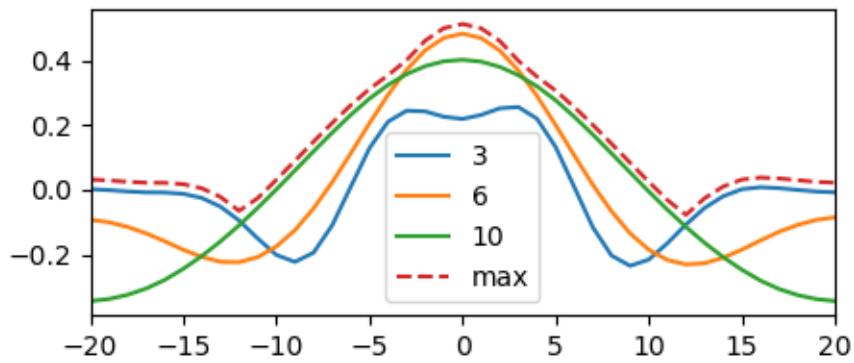
The final step is to look for local extrema on these time series: local maxima (in case of C_{max}) and local minima (in case of C_{min}). It is worth thresholding the generated local extrema to keep only significant detections. The detections can be further filtered by the value of Σ_{min} or Σ_{max} if we have a preliminary idea of the range of the width of the ridge.



(a) Input signal with an angular ridge with a width of 12.

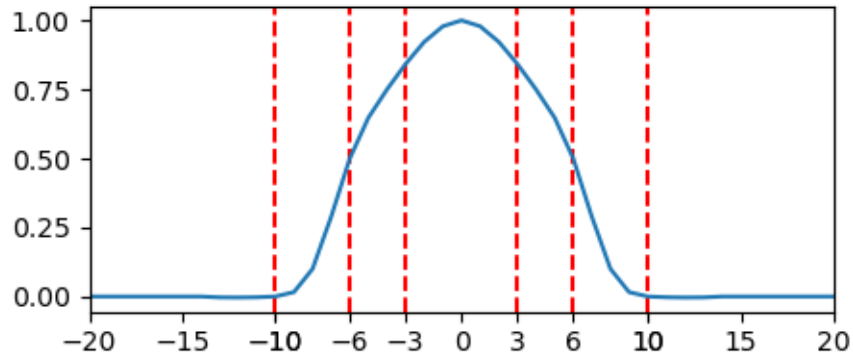


(b) Normalized Gaussian second derivative kernels with different σ s.

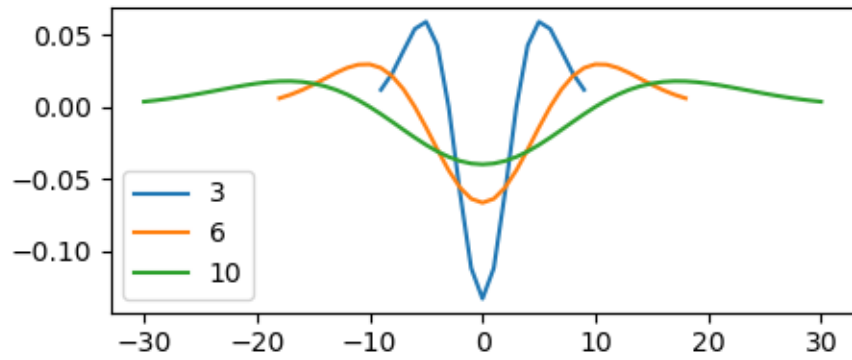


(c) Result of convolving the input signal with kernels of different σ .

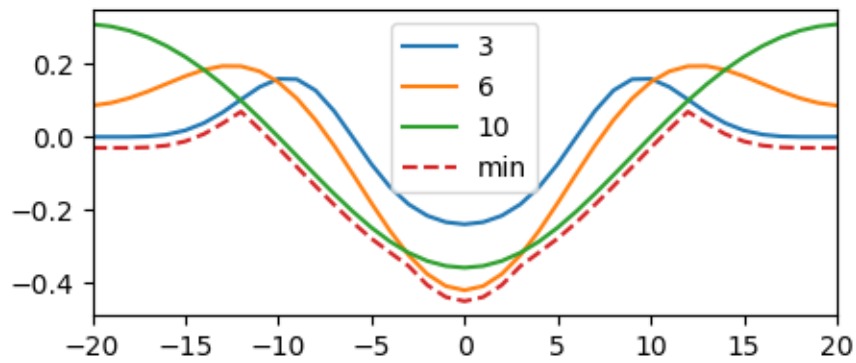
Figure 3.17: Example 1 of convolving a signal with normalized second derivative of Gaussian kernels of different σ .



(a) Input signal with a round ridge having a full width at half maximum of 12.



(b) Normalized Gaussian second derivative kernels with different σ s.



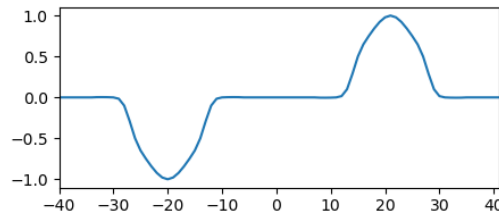
(c) Result of convolving the input signal with kernels of different σ .

Figure 3.18: Example 2 of convolving a signal with normalized second derivative of Gaussian kernels of different σ .

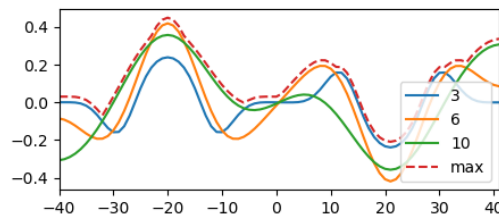
3.5.2.2 Application

To detect lip compression the ridge detector described above was utilized. During the compression, the lips thin into a line which is also noticeable on the lip thickness time series as a "down" ridge. After detecting these ridges and investigating the results, some false detections due to the movement of the head could be observed. These were detected with the same detector based on the "pitch" angle of the head (from Section 3.3.1), and then filtered out.

Although many lip compressions were detected using this method, false detections and undetected cases remained. The remaining false detections were mainly due to an increase in the detected lip size during speech. A weakness of this approach is that due to the nature of the method in such a case a large- σ "down" ridge at the beginning of a large "up" ridge is detected. This phenomenon can be observed in Figure 3.19b: there are salient values at around 10, 30 and 40. This problem couldn't be solved by simply adjusting the threshold parameter of the local extrema detection.



(a) Input signal with a "down" and an "up" ridge, both having a full width at half maximum of 12.

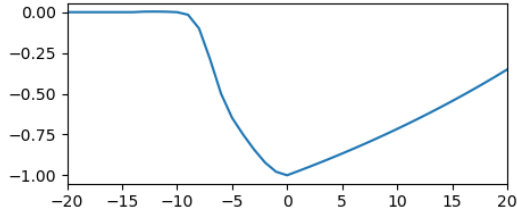


(b) Result of convolving the input signal with kernels of different σ . A "true" peak can be observed at -20 and several "false" peaks at 10, 30 and 40.

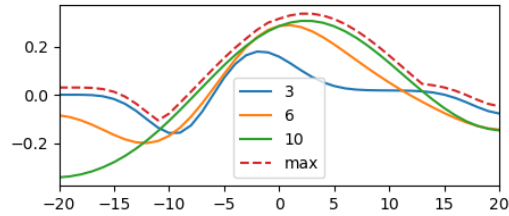
Figure 3.19: Example of convolving a "down" and an "up" ridge with normalized second derivative of Gaussian kernels of different σ .

There are two observable types of forms of the lip compression on the time series: a short ridge or a chasm with a long-term upward trend (like in Figure 3.20a). Another disadvantage of this approach, that the second type of forms (asymmetric chasms) have a much weaker response to the convolution than the ridges. This problem can be observed in Figure 3.20b, where the response with the highest absolute value is at around 0.3, while the highest absolute value in case of a ridge with the same height in Figure 3.18c is around 0.4. This again makes it difficult to threshold the local extrema.

It also contributes that the response of flat chasms (like in Figure 3.21) resulting from the sudden movement of the head may be of similar strength (e.g. 2.5 in Figure 3.21b).

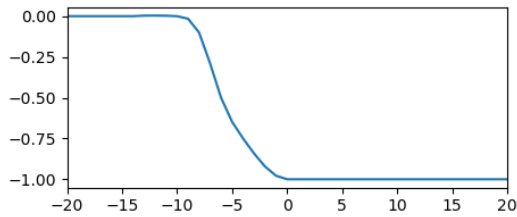


(a) Input signal with a chasm and a long-term upward trend.

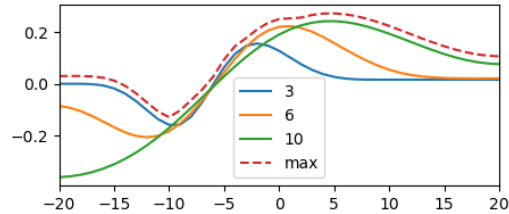


(b) Result of convolving the input signal with kernels of different σ .

Figure 3.20: Example of convolving a chasm long-term upward trend with normalized second derivative of Gaussian kernels of different σ .



(a) Input signal with a chasm.



(b) Result of convolving the input signal with kernels of different σ .

Figure 3.21: Example of convolving a chasm with normalized second derivative of Gaussian kernels of different σ .

3.5.3 Chasm detection-based lip compression detection

To solve the two previously described problems of the implemented ridge detector, a modified version version of it was designed.

3.5.3.1 Chasm detection

The convolution step described in Section 3.5.2.1 can be modified so that at a given point, the convolution is not calculated in the conventional way, but the signal is reflected symmetrically to the point. This can be formulated as:

$$\begin{aligned}
C_{R,\sigma}(t) &= (y' * \widehat{G''_{\sigma}})(t), \quad \text{where } y'(x) = \begin{cases} y(x) & \text{if } x \leq t \\ y(2t - x) & \text{if } x > t \end{cases} \\
&= \int_{x=0-0}^{\infty} y(t-x) \widehat{G''_{\sigma}}(x) dx + \int_{x=-\infty}^{0-0} y(t+x) \widehat{G''_{\sigma}}(x) dx \quad / x' := -x \\
&= \int_{x=0-0}^{\infty} y(t-x) \widehat{G''_{\sigma}}(x) dx - \int_{x'=\infty}^{0+0} y(t-x') \widehat{G''_{\sigma}}(-x') dx' \\
&= \int_{x=-\infty}^{\infty} y(t-x) \widehat{G''_{R,\sigma}}(x) dx, \quad \text{where } \widehat{G''_{R,\sigma}}(x) = \begin{cases} \widehat{G''_{\sigma}}(x) & \text{if } x = 0 \\ 2 \cdot \widehat{G''_{\sigma}}(x) & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}
\end{aligned}$$

This modification can also be derived for mirroring in the other direction:

$$C_{L,\sigma}(x) = \int_{x=-\infty}^{\infty} y(t-x) \widehat{G''_{L,\sigma}}(x) dx, \quad \text{where } \widehat{G''_{L,\sigma}}(x) = \begin{cases} \widehat{G''_{\sigma}}(x) & \text{if } x = 0 \\ 0 & \text{if } x > 0 \\ 2 \cdot \widehat{G''_{\sigma}}(x) & \text{if } x < 0 \end{cases}$$

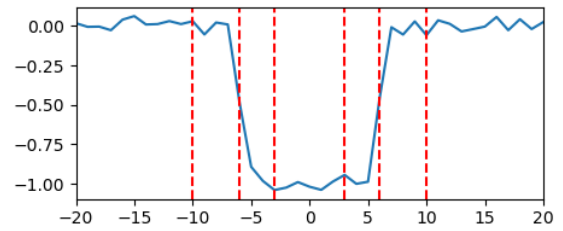
The resulting convolved results can be combined considering the direction to be detected (i.e. to detect "up" or "down" ridges/chasms). The appropriate σ values can also be selected. In case of a ridge, $C_{L,\sigma}$ and $C_{R,\sigma}$ (with an adequate sigma value) have a high absolute value with the same sign in the center of a the ridge as can be seen in Figure 3.22b and Figure 3.23b. Furthermore, the "false" salient values have the opposite sign: see yellow ($\sigma = 6$) values in Figure 3.22b at -13 (top) and 13 (bottom) or in Figure 3.23b at -13 (top) and 13 (bottom). It can also be observed that at these points the value is around 0 for the convolution with the mirrored kernel. This means, that taking the element-wise minimum (or maximum) of $C_{L,\sigma}$ and $C_{R,\sigma}$ with 0 can eliminate these parts. Element-wise minimum is taken in case of "up" and maximum in case of "down" ridge/chasm detection. Afterwards, taking the element-wise geometric mean of the two signals, convolutional values with high absolute value and same sign are preserved, while other values are zeroed out or weighted down as Figure 3.22c and Figure 3.23c show. (Taking the element-wise positive square-root plays a re-normalizing role here.)

$$\begin{aligned}
C_{\text{up}}(x) &= \max_{\sigma} \sqrt{\min \{C_{L,\sigma}(x), 0\} \cdot \min \{C_{R,\sigma}(x), 0\}} \\
C_{\text{down}}(x) &= \max_{\sigma} \sqrt{\max \{C_{L,\sigma}(x), 0\} \cdot \max \{C_{R,\sigma}(x), 0\}}
\end{aligned}$$

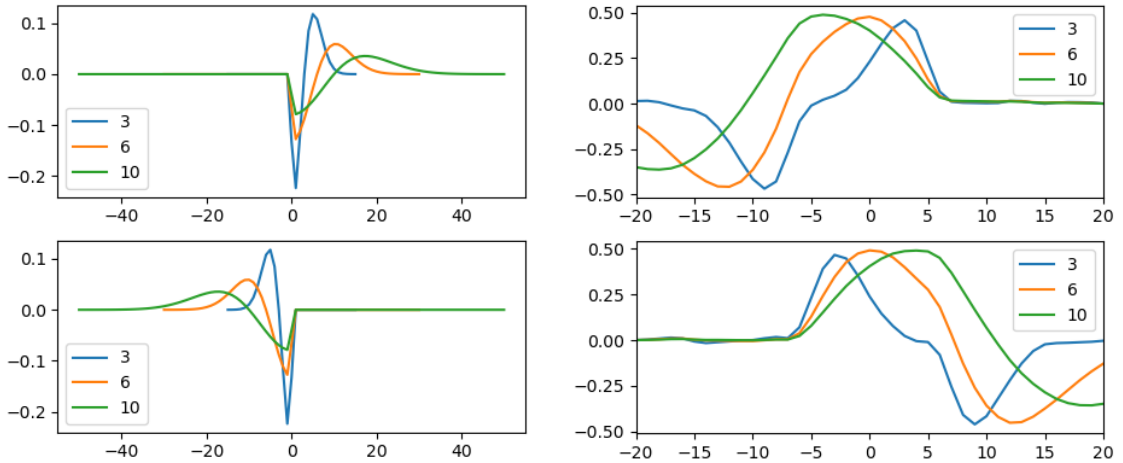
The final step is to look for local maxima over a defined threshold. The final detections can be filtered by width: it can be inferred from the sigma here as well (as described in Section 3.5.2.1).

$$\Sigma_{\text{up}}(x) = \arg \max_{\sigma} \sqrt{\min \{C_{L,\sigma}(x), 0\} \cdot \min \{C_{R,\sigma}(x), 0\}}$$

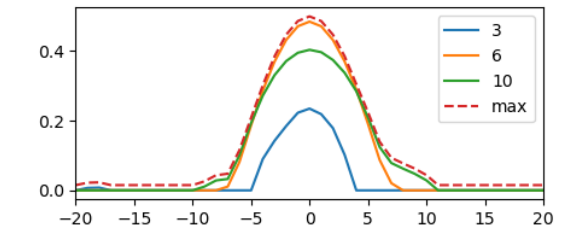
$$\Sigma_{\text{down}}(x) = \arg \max_{\sigma} \sqrt{\max \{C_{L,\sigma}(x), 0\} \cdot \max \{C_{R,\sigma}(x), 0\}}$$



(a) Input signal with an angular ridge having a full width at half maximum of 12.



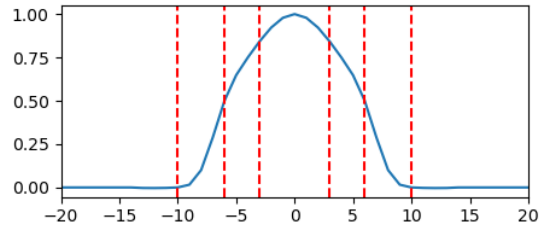
(b) Normalized Gaussian second derivative half kernels with different σ s and the corresponding convolution results.



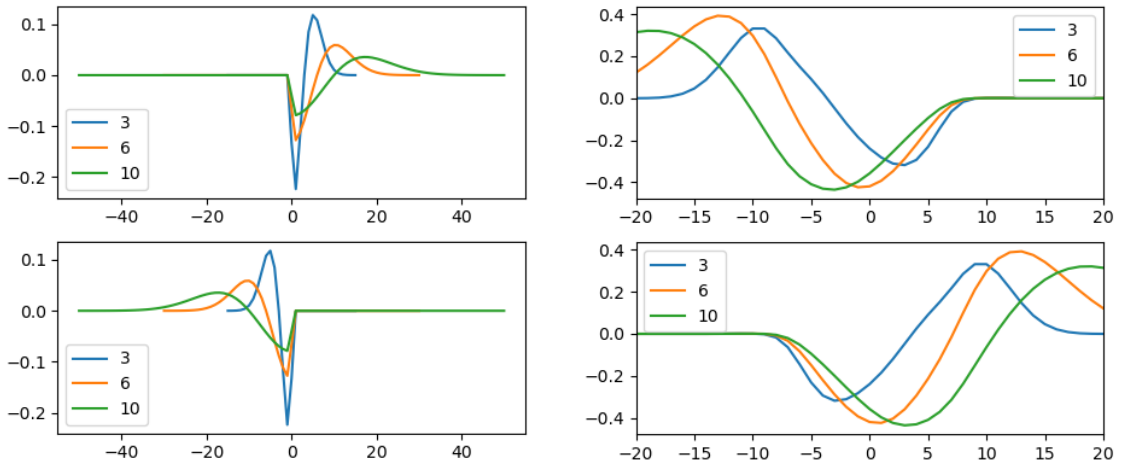
(c) Combination of the convolution results.

Figure 3.22: Example 1 for convolving a signal with normalized second derivative of Gaussian half kernels of different σ .

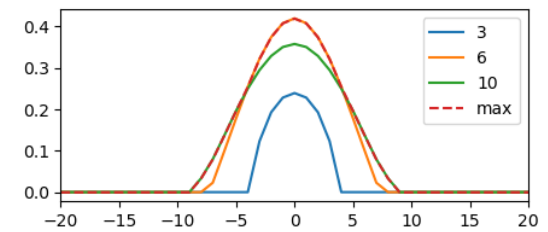
Figure 3.24c shows an example of how chasm detector works in case of an "up" chasm. As can be seen, the response is weak, therefore it doesn't cause false detections. Figure 3.25 demonstrates the difference between the signals produced by the two detectors at which



(a) Input signal with a round ridge having a full width at half maximum of 12.



(b) Normalized Gaussian second derivative half kernels with different σ s and the corresponding convolution results.



(c) Combination of the convolution results.

Figure 3.23: Example 2 of convolving a signal with normalized second derivative of Gaussian half kernels of different σ .

an local extrema are to be detected. The resulting signals are less "noisy" in case of the chasm detector (bottom). Also, the local extrema are easier to threshold.

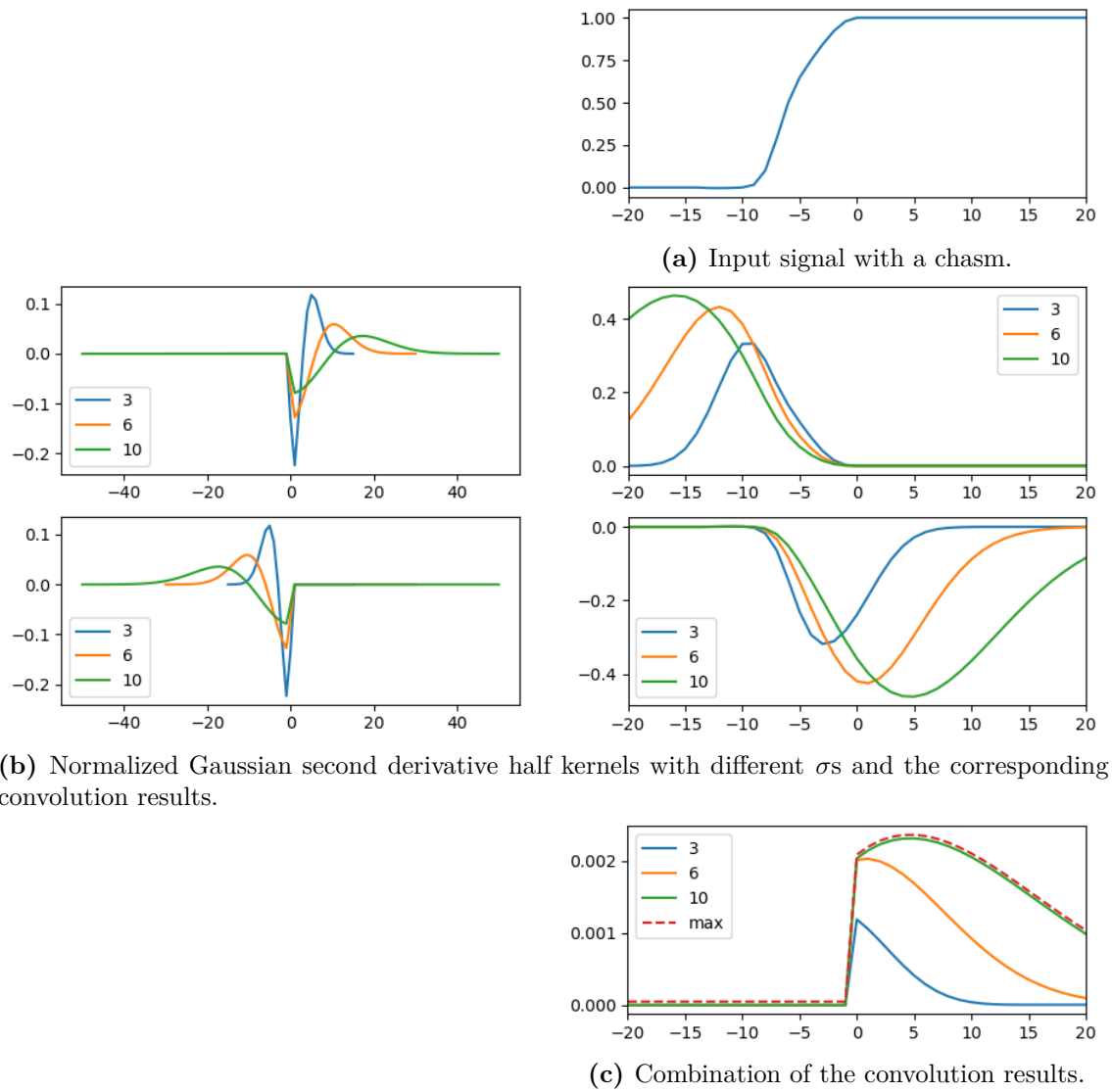
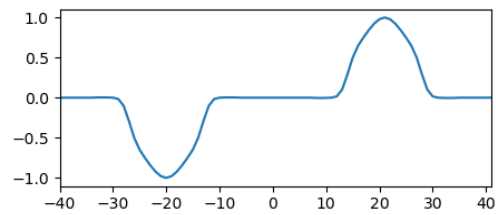
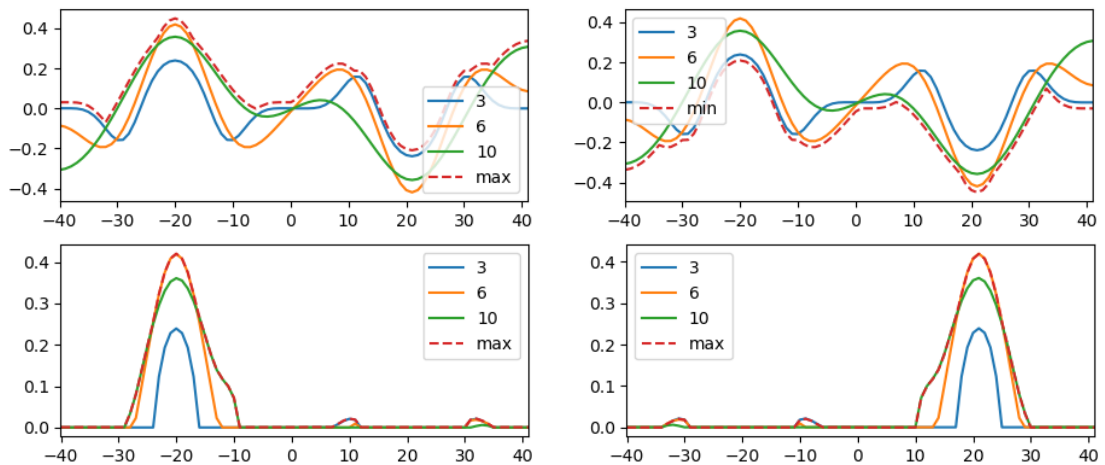


Figure 3.24: Example 3 of convolving a signal with normalized second derivative of Gaussian half kernels of different σ .

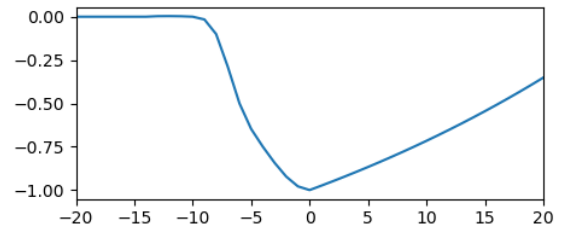


(a) Input signal with a "down" and an "up" ridge, both having a full width at half maximum of 12.

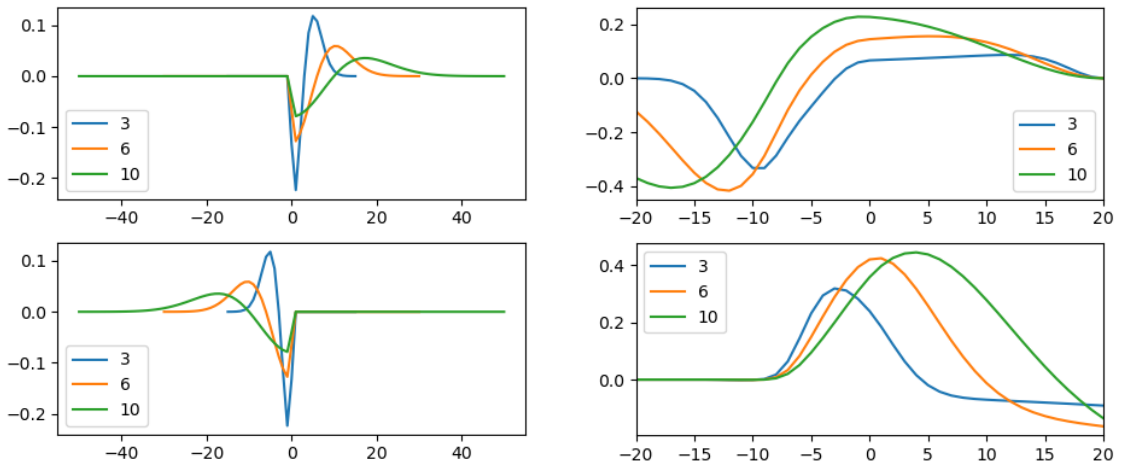


(b) Result of convolving the input signal with kernels of different σ in case of the ridge detector (up) and the chasm detector (down). The signal to be thresholded for "down" (left) and "up" (right) ridge detection is marked with dashed red line. It is easier to threshold in case of the chasm detector.

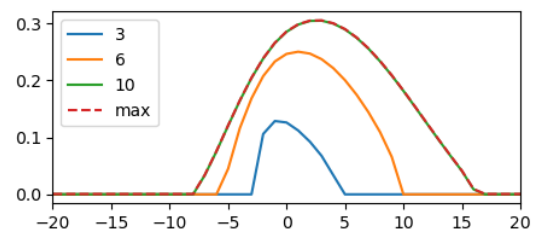
Figure 3.25: Example to demonstrate the advantage of the chasm detection in case of "down" and "up" ridges.



(a) Input signal with a chasm and a long-term increasing trend.



(b) Normalized Gaussian second derivative half kernels with different σ s and the corresponding convolution results.

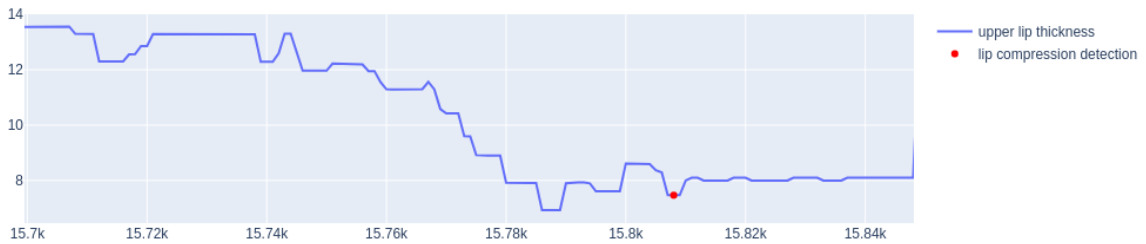


(c) Combination of the convolution results.

Figure 3.26: Example to demonstrate the advantage of the chasm detection in case of a chasm with a long-term increasing trend.

3.5.3.2 Application

After redesigning the detection method the chasm detector was utilized to detect lip compression. This detector handled cases better where the lip suddenly thinned out and then slowly recovered in thickness. Furthermore, many false detections were eliminated due to the "mirroring" effect of the chasm detector. The further operation of the detection procedure is unchanged, thus false detections coming from nodding are still filtered out. Figure 3.27 and Figure 3.28 show examples of the lip compression.

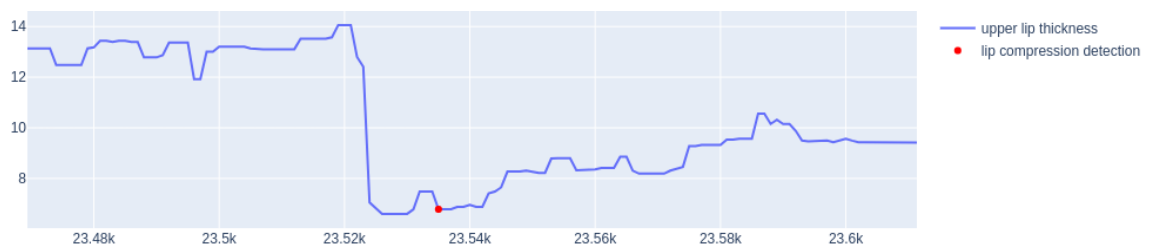


(a) Lip thickness time series. A red marker marks the detected lip compression where the upper lip has thinned.



(b) Frames before lip compression and at the moment of the detection. Source: YouTube [64]

Figure 3.27: Example 1 of detecting a lip compression.



(a) Lip thickness time series. A red marker marks the detected lip compression where the upper lip has thinned.



23510

23535

(b) Frames before lip compression and at the moment of the detection. Source: YouTube [64]

Figure 3.28: Example 2 of detecting a lip compression.

Chapter 4

Evaluation

4.1 Contempt detection

We evaluated our contempt detection solution on the subset of the BAUM-1s data set. We kept the videos of the firsts 12 subjects as they were re-annotated by our psychologist. We had 45 video samples annotated with and 296 video clips without contempt expression.

We detected the strength of the nasolabial fold on each side of the face and for each frame of the video. Our algorithm classified a video as one containing contempt expression if the nasolabial fold strength was above a threshold T exactly on one side of the face for at least F frames. We tuned T and F empirically, balancing between specificity and sensitivity values. We decided against using accuracy to measure the performance of our solution because the data set is highly unbalanced. The best T and F values were 500 and 3. With these parameters we achieved an overall 0.71 sensitivity and 0.61 specificity. We also examined the sensitivity and specificity scores for each subject individually as shown in Table 4.1.

Subject	Sensitivity	Specificity
S001	— (0)	0.55 (11)
S002	0.00 (1)	0.39 (36)
S003	1.00 (7)	0.54 (26)
S004	0.82 (11)	0.23 (22)
S006	0.69 (16)	0.72 (18)
S007	0.75 (4)	0.32 (31)
S008	0.67 (3)	0.57 (28)
S009	— (0)	0.88 (40)
S010	0.00 (1)	0.88 (51)
S012	0.00 (2)	0.73 (33)

Table 4.1: Sensitivity and specificity of contempt detection for individual subjects. The number of videos with and without contempt expression are shown in parentheses.

We could well identify the contempt expression for most of the subjects. S002 had only one video labelled with contempt. In that video the nasolabial fold was too weak during the contempt expression for our detector. S010 and S012 were young females and the nasolabial fold did not appear on their video clips.

On the other hand we had high false positive rate for some subjects for multiple reasons. S002 had a mustache that was thick even in the corner of his mouth. The Gabor filter did not highlight the corner of the mustache because it was steeper than 15° , and the mustache removal algorithm could not connect the skeleton fragment that belonged to the corner of the mustache with the rest and therefore it was not removed as shown Figure 4.1.

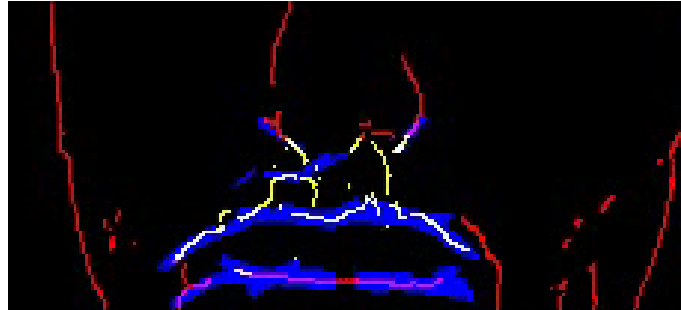


Figure 4.1: The problems of mustache removal: The white and yellow lines shows the parts of the skeleton eliminated in the process $(M^* \cap S)$, on the right side of the image the corner of the mustache is red and will not be removed.

S004 turned his head slightly to the right and tilts his head to the left in most of his videos. Which caused his left nasolabial fold to be more intensive than the right one while he was smiling, was disgusted or was angry (see Figure 4.2). When our algorithm could not detect the weak right nasolabial fold it classified the video incorrectly as contempt.

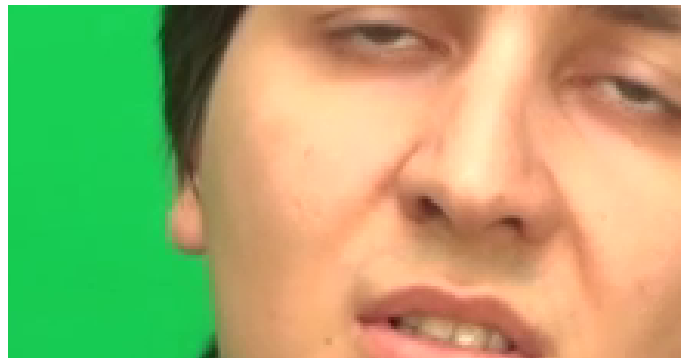


Figure 4.2: S004 tilts his head slightly to the left which caused the disgust expression to be asymmetric: left nasolabial wrinkle is deeper than the right one.

In case of S007, our detector was mistaken mostly because the asymmetric nasolabial fold intensities as well. An other problem during the processing of his video clips was, that his nasolabial fold was connected to the mustache at the corner of his mouth and our mustache elimination method eliminated the nasolabial fold as well, but only one side of his face. This caused the originally symmetric smiling to have asymmetric wrinkle scores (see Figure 4.3).

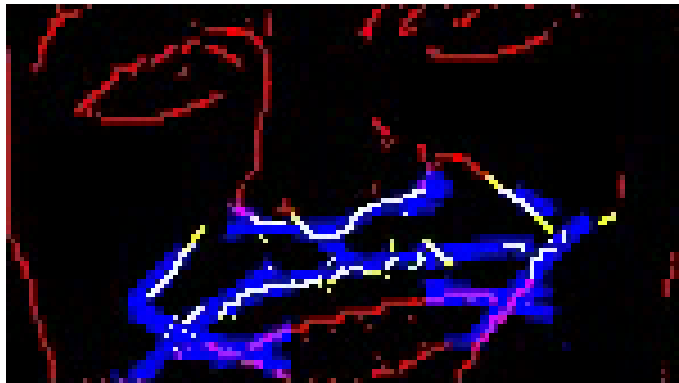
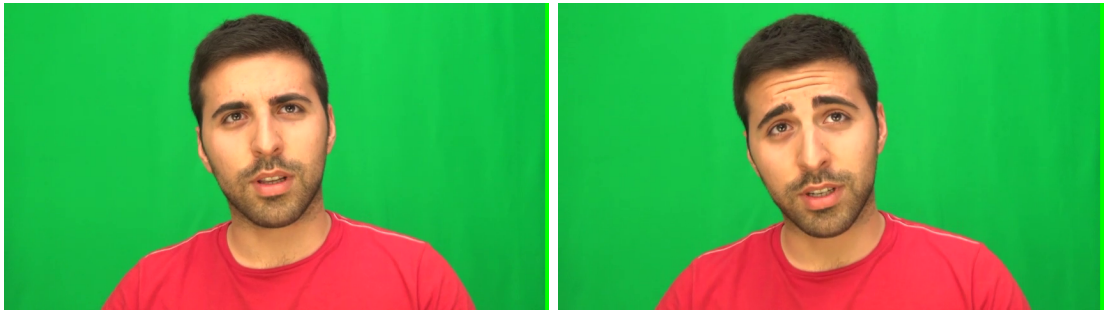


Figure 4.3: S007 smiles. The skeleton fragment which will be emitted are shown with white. The whole left nasolabial fold and the lower part of the right nasolabial are also white, so they will be emitted. Only the upper part of the right nasolabial fold will remain which makes the expression asymmetric

4.2 Nod detection

The evaluation of the nod detection algorithm was performed on manually selected videos from the BAUM dataset. Videos containing visible nods were chosen. The videos were divided into two groups: videos containing clearly visible and hardly noticeable nodding. A total of 28 videos containing large nods and 32 containing small nods were selected for evaluation. In the evaluation process, subsequent nods were treated as one and considered recognized if one of them was found. A detection is considered false if it cannot be linked to any nodding movement. 60 nods were observed in videos with a large nod from which 35 (58.3%) were detected and there were 5 false detections. In case of the videos containing small nods, 41 nods could be observed, from which 9 (21.95%) were detected and 6 were detected falsely. The causes of false negatives and false positive are detailed below:

- **Slight nod.** Although videos containing visible 'pitch' movement were selected, some of them were not visible in the time series or the jump in the 'pitch' values was under the threshold.
- **Subsequent nods.** Subsequent nods tend to have a smaller amplitude thus not even a part of it can be detected.
- **Head tilted sideways.** If the head is tilted in the 'roll' angle, a nod also induces a change in the 'yaw' angle. These detections are filtered out by the algorithm, in this case falsely. An example to this phenomenon is shown in Figure 4.4a.
- **Gestures similar to nodding (for the algorithm).** Since the main input of the detection is the angle of the head, gestures like straightening the back or throwing back the head, look similar to a nod in the time series.



(a) Nod while the head is tilted sideways.

Figure 4.4: An example of false negative nod detection. Source: BAUM dataset [70]

4.3 Eyebrow movement

To evaluate the eyebrow raising, videos containing eyebrow raising were selected from the BAUM dataset. The videos were selected simply on the basis of whether they showed eyebrow raising, so videos with both slight and clearly visible raisings were selected. The detection was performed on a total of 54 videos. These videos contained 75 eyebrow raisings from which 57 (76%) was detected. In addition, there were 6 false positive detections. The causes of false negatives were investigated when we reviewed results. These are detailed below:

- **Slight eyebrow raising.** The jump in values is below the threshold or the movement is not even visible on the eyebrow-eye distance time series.
- **One-sided eyebrow raising.** Usually, in such cases the landmark detector is not accurate enough (such a sample is rare in its training set) and the displacement of the eyebrows on the raised side is less observable than if they were raised on both sides. Such a case is shown in Figure 4.5a. It makes the detection more difficult, if it is a slight eyebrow raise as shown in Figure 4.5b.
- **The detection is filtered out due to shake detection.** This is an error caused by the characteristics of the shake detection algorithm: if the person in the video moves their hand next to their head (as shown in Figure 4.5c), it is detected as a camera shake, thus the detection is filtered out.
- **The eyebrows take too long to relax.** This is due to the characteristics of the pattern detection algorithm: eyebrow movements with a length above a predefined threshold are filtered out.
- **Nod during an eyebrow raising.** Although many false positives can be eliminated by filtering out possibly fake detections during a nod, some of the true detections are also discarded. Figure 4.5d shows an example of this phenomenon.



(a) One-sided eyebrow raising.



(b) A slight one-sided eyebrow raising.



(c) Hands next to the head: the camera shake detector gives a false signal, thus the eyebrow detection is filtered out.



(d) A nod during the eyebrow raising: the detected eyebrow raising is filtered out.

Figure 4.5: Examples of false negative eyebrow raising detections.
Source: BAUM dataset [70]

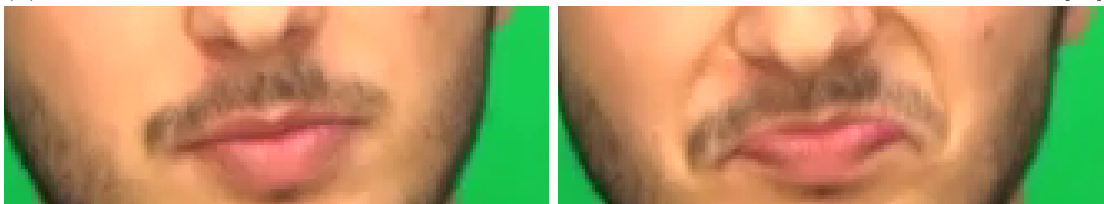
4.4 Lip compression

Unfortunately, no proper evaluation of the lip compression detector could be performed. Lip compression is a rare microexpression and we could not even find an unlabeled dataset for the evaluation. Based on the few videos the algorithm could be tested on, the experiences are the following:

- The detector performs well if the lip of the person in the video is clearly visible, i.e. the contour of the upper lip can be easily determined by a human.
- If the lip is pale in color and not sharply differentiated from the color of the skin above, the output of the lip thickness algorithm becomes noisy, resulting in false positive detections.
- If the person in the video has a mustache, the lip thickness estimation algorithm makes mistakes often. This is because there is usually a greater gradient at the edge of the moustache than at the edge of the upper lip, and in these cases, the upper part of the landmark based lip segmentation usually identifies the bottom of the mustache.



(a) Pale colored lips. The contour of the upper lip is not clearly visible. Source: YouTube [64]



(b) A person with a mustache. The lip thickness estimator makes more mistakes in this case. Source: [70]

Figure 4.6: Examples of difficult lip compression detection cases.

Chapter 5

Discussion

In our work, we introduced several hybrid intelligent-expert image processing algorithms for facial expression and gesture detection. We proposed solutions for contempt, nod, eyebrow movement, and lip compression detection. Our implemented methods perform well in case of unambiguous expressions and gestures. However, there is still much room for future improvements, both in terms of eliminating false positives and false negatives. False positives are mainly caused by the complexity of the facial expressions, the diversity of the faces, and the limited robustness of our algorithms.

Expert knowledge-based algorithms utilize expert-designed models. These models balance between simplicity and accuracy. Most of the false detections occur in such cases when our model utilizes larger approximations and relaxations than what would be adequate for correct detection. We are going to modify the algorithms to handle these cases during our forthcoming research in this area.

On the other hand, we aimed to detect microexpressions in real-life scenarios which involve recordings of various quality, i.e. different lighting conditions and varying resolutions. Furthermore, recordings were occasionally made by non-fixed cameras, thus motion related noise had to be taken into consideration. In other words, handling all these conditions are challenging, and most conventional solutions aim to alleviate only a portion of these. Nonetheless, we aimed to adapt our methods to a wide range of conditions, and provided reasonable results.

An alternative solution to these problems is to utilize adaptive algorithms, e.g. machine learning, but the low number of adequate training samples can be a bottleneck. Therefore, native machine learning-based solutions cannot be utilized for this purpose, a hybrid system must be used instead. In these solutions, the number of required samples can be decreased by injecting prior information based elements into the learning system. This approach could also be explored as an option for further development in the future.

In case of every examined gesture, our task was mainly the algorithmization of the domain specific knowledge, which proved to be challenging. However, these algorithm-based solutions scale very well with the size of the sample set used in their construction. Our algorithms perform reasonably well on problems where there are not enough samples to properly evaluate, let alone train a fully adaptive machine learning-based system. According to an expert psychologist, in its current form it can be a useful tool for behavioral analysis.

Acknowledgements

We would like to thank Dániel Hadházi and Gábor Hullám for their help and advice during the preparation of this thesis and the related research. We also want to thank our psychologist expert in microexpressions for her advice and for the annotation of the dataset. The research of Gábor Hullám was supported by the ÚNKP-21-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund, and the János Bolyai research scholarship.

Bibliography

- [1] Youssef Ibrahim Abdel-Aziz. *PHOTOGRAMMETRIC POTENTIAL OF NON-METRIC CAMERAS*. University of Illinois at Urbana-Champaign, 1974.
- [2] Apple ARKit. Apple arkit. <https://developer.apple.com/documentation/arkit/>, 2017. Accessed: 2021-05-05.
- [3] R.R. Avent, Chong Teck Ng, and J.A. Neal. Machine vision recognition of facial affect using backpropagation neural networks. In *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 1364–1365 vol.2, 1994. DOI: 10.1109/IEMBS.1994.415474.
- [4] Xianye Ben, Yi Ren, Junping Zhang, Su-Jing Wang, Kidiyo Kpalma, Weixiao Meng, and Yong-Jin Liu. Video-based facial micro-expression analysis: A survey of datasets, features and algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [5] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [6] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.
- [7] Yiqiang Chen, Yu Yu, and Jean-Marc Odobez. Head nod detection from a full 3d model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 136–144, 2015.
- [8] Pengyu Cong, Zhiwei Xiong, Yueyi Zhang, Shenghui Zhao, and Feng Wu. Accurate dynamic 3d sensing with fourier-assisted phase shifting. *IEEE Journal of Selected Topics in Signal Processing*, 9(3):396–408, 2014.
- [9] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision — ECCV’98*, pages 484–498, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69235-5.
- [10] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.
- [11] Jian S Dai. Euler–rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 92:144–152, 2015.
- [12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 886–893 vol. 1, 2005. DOI: 10.1109/CVPR.2005.177.

- [13] Adrian K Davison, Cliff Lansley, Nicholas Costen, Kevin Tan, and Moi Hoon Yap. Samm: A spontaneous micro-facial movement dataset. *IEEE transactions on affective computing*, 9(1):116–129, 2016.
- [14] Fernando De la Torre, Wen-Sheng Chu, Xuehan Xiong, Francisco Vicente, Xiaoyu Ding, and Jeffrey Cohn. Intraface. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–8, 2015. DOI: 10.1109/FG.2015.7163082.
- [15] Paul Ekman and Karl G. Heider. The universality of a contempt expression: A replication. *Motivation and Emotion*, 12(3):303–308, Sep 1988. ISSN 1573-6644. DOI: 10.1007/BF00993116. URL <https://doi.org/10.1007/BF00993116>.
- [16] Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.
- [17] Paul Ekman, Joseph C. Hager, and Wallace V. Friesen. *Facial action coding system: the manual*. Research Nexus, 2002.
- [18] Ro Frangi, W.J. Niessen, Koen Vincken, and Max Viergever. Multiscale vessel enhancement filtering. *Med. Image Comput. Comput. Assist. Interv.*, 1496, 02 2000.
- [19] Balazs Frey and Gabor Revy. Microexpression detection using hybrid expert systems. unpublished, 2020.
- [20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.
- [21] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93, Part III(26):429–457, November 1946.
- [22] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.
- [23] Jan Gorisch, Corine Astésano, Ellen Gurman Bard, Brigitte Bigi, and Laurent Prévot. Aix map task corpus: The french multimodal corpus of task-oriented dialogue. In *9th International conference on Language Resources and Evaluation*, pages 2648–2652, 2014.
- [24] Xiaojie Guo, Siyuan Li, Jinke Yu, Jiawan Zhang, Jiayi Ma, Lin Ma, Wei Liu, and Haibin Ling. Pfl: A practical facial landmark detector. *arXiv e-prints*, pages arXiv–1902, 2019.
- [25] Diyar Gür, Niklas Schäfer, Mario Kupnik, and Philipp Beckerle. A human–computer interface replacing mouse and keyboard for individuals with limited upper limb mobility. *Multimodal Technologies and Interaction*, 4(4):84, 2020.
- [26] Jihun Hamm, Christian G Kohler, Ruben C Gur, and Ragini Verma. Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. *Journal of neuroscience methods*, 200(2):237–256, 2011.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.

- [28] Petr Husák, Jan Cech, and Jiří Matas. Spotting facial micro-expressions “in the wild”. In *22nd Computer Vision Winter Workshop (Retz)*, 2017.
- [29] Laurent Prévot Jan Gorisch. Aix-dvd, 2014. URL <https://hdl.handle.net/11403/sldr000891/v1>. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.
- [30] Paula Jones, Paul Viola, and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *University of Rochester. Charles Rich*. Citeseer, 2001.
- [31] Miyuki Kamachi, Michael Lyons, and Jiro Gyoba. The japanese female facial expression (jaffe) database. Available: <http://www.kasrl.org/jaffe.html>, 01 1997.
- [32] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.
- [33] Fuzail Khan. Facial expression recognition using facial landmark detection and feature extraction via neural networks. *arXiv preprint arXiv:1812.04510*, 2018.
- [34] Gulraiz Khan, Aiman Siddiqi, Muhammad Usman Ghani Khan, Samyan Qayyum Wahla, and Sahar Samyan. Geometric positions and optical flow based emotion detection using mlp and reduced dimensions. *IET Image Processing*, 13(4):634–643, 2019.
- [35] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [36] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML ’01*, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.
- [37] Elmar H Langholz and Reuben Brasher. Real-time on-device nod and shake recognition. *arXiv preprint arXiv:1806.04776*, 2018.
- [38] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.
- [39] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.
- [40] Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao, and Matti Pietikäinen. A spontaneous micro-expression database: Inducement, collection and baseline. In *2013 10th IEEE International Conference and Workshops on Automatic face and gesture recognition (fg)*, pages 1–6. IEEE, 2013.
- [41] Tony Lindeberg. *Scale-Space Theory in Computer Vision*, volume 256. Springer Science & Business Media, 1993.
- [42] Jingjing Liu, Bo Liu, Shaoting Zhang, Fei Yang, Peng Yang, Dimitris N Metaxas, and Carol Neidle. Recognizing eyebrow and periodic head gestures using crfs for non-manual grammatical marker detection in asl. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2013.

- [43] Yinglu Liu, Hao Shen, Yue Si, Xiaobo Wang, Xiangyu Zhu, Hailin Shi, Zhibin Hong, Hanqi Guo, Ziyuan Guo, Yanqin Chen, et al. Grand challenge of 106-point facial landmark localization. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 613–616. IEEE, 2019.
- [44] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, 2010. DOI: 10.1109/CVPRW.2010.5543262.
- [45] Daniel Lundqvist, Anders Flykt, and Arne Öhman. The karolinska directed emotional faces (kdef). *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, 91(630):2–2, 1998.
- [46] Jiri Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78: 119–137, 04 2000. DOI: 10.1006/cviu.1999.0831.
- [47] Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, et al. The ami meeting corpus. In *Proceedings of the 5th international conference on methods and techniques in behavioral research*, volume 88, page 100. Citeseer, 2005.
- [48] Choon-Ching Ng, Moi Hoon Yap, Nicholas Costen, and Baihua Li. Automatic wrinkle detection using hybrid hessian filter. In *Asian Conference on Computer Vision*, pages 609–622. Springer, 2014.
- [49] Choon-Ching Ng, Moi Hoon Yap, Nicholas Costen, and Baihua Li. Wrinkle detection using hessian line tracking. *IEEE Access*, 3:1079–1088, 2015. DOI: 10.1109/ACCESS.2015.2455871.
- [50] Catharine Oertel, Kenneth A Funes Mora, Samira Sheikhi, Jean-Marc Odobez, and Joakim Gustafson. Who will get the grant? a multimodal corpus for the analysis of conversational behaviours in group interviews. In *Proceedings of the 2014 Workshop on Understanding and Modeling Multiparty, Multimodal Interactions*, pages 27–32, 2014.
- [51] Sébastien Ouellet. Real-time emotion recognition for gaming using deep convolutional network features. *arXiv*, pages arXiv–1408, 2014.
- [52] Fangbing Qu, Su-Jing Wang, Wen-Jing Yan, He Li, Shuhang Wu, and Xiaolan Fu. Cas (me)²: a database for spontaneous macro-expression and micro-expression spotting and recognition. *IEEE Transactions on Affective Computing*, 9(4):424–436, 2017.
- [53] Stéphane Rauzy and Aurélie Goujon. Automatic annotation of facial actions from a video record: The case of eyebrows raising and frowning. In *Workshop on "Affects, Compagnons Artificiels et Interactions", WACAI 2018*, pages 7–pages, 2018.
- [54] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [55] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016.

- [56] Mark Sandler, Andrew G Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.
- [57] Thibaud S en echal, Jay Turcot, and Rana El Kaliouby. Smile or smirk? automatic detection of spontaneous asymmetric smiles to understand viewer experience. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–8. IEEE, 2013.
- [58] Wenzhao Tan and Gang Rong. A real-time head nod and shake detector using hmms. *Expert Systems with Applications*, 25(3):461–466, 2003.
- [59] Dhananjay Thekkedath and R. R. Sedamkar. Detecting affect states using vgg16, resnet50 and se-resnet50 networks. *SN Computer Science*, 1(2):79, Mar 2020. ISSN 2661-8907. DOI: 10.1007/s42979-020-0114-9. URL <https://doi.org/10.1007/s42979-020-0114-9>.
- [60] Jessica L Tracy and Richard W Robins. The automaticity of emotion recognition. *Emotion*, 8(1):81, 2008.
- [61] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001.
- [62] Peng Wang, Frederick Barrett, Elizabeth Martin, Marina Milonova, Raquel E Gur, Ruben C Gur, Christian Kohler, and Ragini Verma. Automated video-based facial expression analysis of neuropsychiatric disorders. *Journal of neuroscience methods*, 168(1):224–238, 2008.
- [63] Haolin Wei, Patricia Scanlon, Yingbo Li, David S Monaghan, and Noel E O’Connor. Real-time head nod and shake detection for continuous human affect recognition. In *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4. IEEE, 2013.
- [64] Patryk Wezowski. Full micro expressions analysis in 4k lie to me style - micro expressions training as in lie to me, 2017. URL <https://www.youtube.com/watch?v=B0tFjWNYRkA>.
- [65] Weicheng Xie, Linlin Shen, and Jianmin Jiang. A novel transient wrinkle detection algorithm and its application for expression synthesis. *IEEE Transactions on Multimedia*, 19(2):279–292, 2017. DOI: 10.1109/TMM.2016.2614429.
- [66] Jingwei Yan, Wenming Zheng, Zhen Cui, Chuangao Tang, Tong Zhang, and Yuan Zong. Multi-cue fusion for emotion recognition in the wild. *Neurocomputing*, 309: 27–35, 2018.
- [67] Wen-Jing Yan, Qi Wu, Yong-Jin Liu, Su-Jing Wang, and Xiaolan Fu. Casme database: A dataset of spontaneous micro-expressions collected from neutralized faces. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–7. IEEE, 2013.
- [68] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, Yu-Hsin Chen, and Xiaolan Fu. Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. *PloS one*, 9(1):e86041, 2014.
- [69] Delina Beh Mei Yin, Shariman Omar, Bazilah A Talip, Amalia Muklas, Nur Afiqah Mohd Norain, and Abu Talib Othman. Fusion of face recognition and facial expression detection for authentication: a proposed model. In *Proceedings of the*

11th International Conference on Ubiquitous Information Management and Communication, pages 1–8, 2017.

- [70] Sara Zhalehpour, Onur Onder, Zahid Akhtar, and Cigdem Eroglu Erdem. Baum-1: A spontaneous audio-visual face database of affective and mental states. *IEEE Transactions on Affective Computing*, 8(3):300–313, 2016.