6-23-2018

Automatic Classification and Shift Detection of Facial Expressions in Event-Aware Smart Environments

Arne Bernin

Larissa Müller

Sobin Ghose

Christos Grecos

Qi Wang

See next page for additional authors

Authors

Arne Bernin, Larissa Müller, Sobin Ghose, Christos Grecos, Qi Wang, Ralf Jettke, Kai von Luck, and Florian Vogt

Automatic Classification and Shift Detection of Facial Expressions in Event-Aware Smart Environments

Arne Bernin
Hamburg University of Applied Sciences
University of the West of Scotland
ab@emotionbike.org

Larissa Müller
Hamburg University of Applied Sciences
University of the West of Scotland
lm@emotionbike.org

Sobin Ghose
Hamburg University of Applied Sciences
Faculty TI
Hamburg, Germany
sg@emotionbike.org

Christos Grecos
Computer Science Department
Central Washington University
Ellensburg, USA

Qi Wang
School of Engineering and Computing
University of the West of Scotland

Ralf Jettke
Faculty of Psychology and Human Movement
University of Hamburg
Hamburg, Germany

Kai von Luck
Department Informatik
Hamburg University of Applied Sciences
Hamburg, Germany

Florian Vogt
Innovations Kontakt Stelle (IKS) Hamburg, Germany
HAW Hamburg
vogt@iks-hamburg.de

ABSTRACT

Affective application developers often face a challenge in integrating the output of facial expression recognition (FER) software in interactive systems: although many algorithms have been proposed for FER, integrating the results of these algorithms into applications remains difficult. Due to inter- and within-subject variations further post-processing is needed. Our work addresses this problem by introducing and comparing three post-processing classification algorithms for FER output applied to an event-based interaction scheme to pinpoint the affective context within a time window. Our comparison is based on earlier published experiments with an interactive cycling simulation in which participants were provoked with game elements and their facial expression responses were analysed by all three algorithms with a human observer as reference. The three post-processing algorithms we investigate are mean fixed-window, matched filter, and Bayesian changepoint detection. In addition, we introduce a novel method for detecting fast transition of facial expressions, which we call *emotional shift*. The proposed detection pattern is suitable for affective applications especially in smart environments, wherever users' reactions can be tied to events.

PETRA '18 June 23–26, 2018, Corfu Island, Greece

DOI: https://doi.org/10.1145/3197768.3201527

CCS Concepts

•Computing methodologies → Activity recognition and understanding; •Human-centered computing → Human computer interaction (HCI);

Keywords

Affective Computing, Facial Expression Recognition, Emotional shift, Emotion transition, Krippendorff's alpha

1. INTRODUCTION

In recent years, affective computing has developed into a vibrant, multi-disciplinary field of research with exciting opportunities for new applications. Its foundations are in emotion models, sensing technologies, affective and social signal processing, affective data sets and reference applications. Despite much progress over two decades many challenges to build working systems remain [28, 26, 6]. During our research into affective solutions for exergames, we encountered numerous difficulties building generalised emotion-enriched applications due to the complex nature of emotions, rational and contextual processing, which occupies a significant portion of the human brain.

The common approach of mimicking human processes by collecting vast amounts of emotion data for situational and cultural contexts, experimental settings and subject groups, parametrised with plausibility rules and tuning parameters, has resulted in substantial challenges for AI research.

Over the course of our research comparing facial expression and emotion recognition systems, we identified the issue of application-specific mapping and its automatic interpretation. While facial-expression-derived emotions are a valuable source of information for event and reaction detection in affective-aware applications, they are also difficult and com-

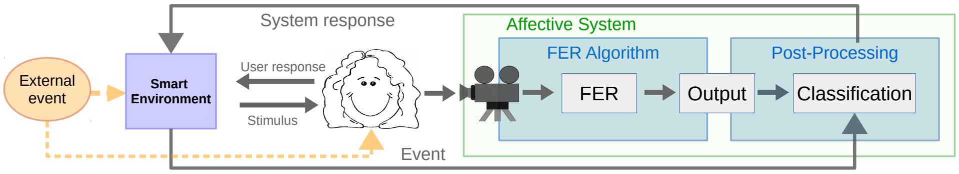


Figure 1: Our pipeline for an event-aware smart affective system. The user's face is recorded by a camera and afterwards analysed with a FER algorithm to extract facial expressions. The algorithms output is post-processed with filtering, classification and interpretation based on external or internal events. This post-processing step is the focus of this publication.

plex to interpret and correlate with user actions, profiles and other data. As a result, FER analysis is quite complex and often not reliable due to response variations (see Fig. 2).

For our work we found it useful to generalise both user actions and internal plus external signals to the application as events. Utilizing FER with these events provided us with additional context. The concept of this study was to extend single frame and average signal approaches to more closely analyse timing and response characteristics.

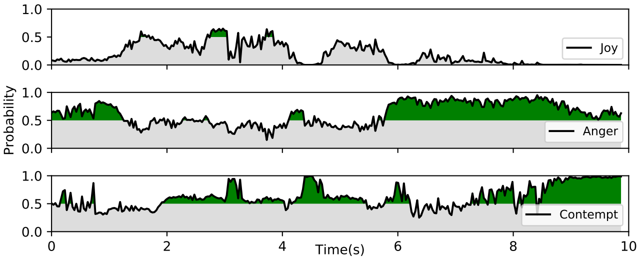


Figure 2: Classification of FER algorithm output can be considered a multi-channel signal processing problem. Often the output of FER algorithms provides one independent channel per expression.

Our general aim is to build robust real functional interactive applications with a variety of users and individual response dynamics, where a one-to-one mapping from expression to reaction is not fixed and post-processing is needed to provide a working system. Our goal is to determine whether the user perceives an event - based on the reaction and emotional expressions. These events may be triggered by system internal provocations, such as audio-visual-haptic stimuli or caused by system external triggers, such as an event from the real world. In the latter case, the system must detect the external event in order to determine the timeframe for the user's reaction as depicted in Fig. 1.

Thus far, we have relied on semi-automatic methods for detecting user reactions [24], as automatic solutions for direct emotion and expression mapping did not work for us.

In this paper we present a comparison of approaches for automatically recognising responses from the output of FER algorithms and provide a benchmark for this post-processing step. We demonstrate our post-processing methods with the state-of-the-art FER system Emotient.

Although this work is based on an emotion provoking exergame, our findings can be applied to any affective scenario in which an application setting provides internal or external events to fix search windows occurring in smart and assistive Environments.

2. RELATED WORK

2.1 Previous Work

In order to understand this work in its context, we find it beneficial to be aware of our previous work, which describes the EmotionBike system [24], presents the experimental setup and showcases the provocation of human reactions with game events analysed post experiment by the facial expression system CERT [17]. The EmotionBike provides an exergame scenario enabling users to cycle through an interactive game on a stationary bike trainer with steering capabilities. It is a variant of a cockpit scenario also suitable for research on the topic of games for physical therapy and orthopaedic rehabilitation. Our follow-up study [23] enhanced the experimental setup with the event-based analysis of galvanic skin response combined with facial expressions. For practical guidance, we presented a benchmark [3] of four automatic facial expression analysis systems with three emotion-labelled reference databases and a systematic method for performance analysis and improvement that allows to tailor for specific application needs.

It is to note that all related work including our own: 1. observe only single frame or short sequence input and 2. use exclusively semi-automatic or manual emotion classification in practical applications that produce high inter- and within-subject variations. One standing research challenge is the fully automatic classification of user reactions, for which we present three alternative algorithms as feasible solutions. Our classification methods can be effectively combined with the application-specific clustering approach [3] to increase its robustness for a wide spectrum of user reactions.

An interesting observation in our previous studies is the way inter- and within subject-responses vary as positive/negative inverse reaction based on predisposition.

2.2 Applications: Smart and Assistive Environments

Smart environments often provide a reasonable application context for recognising emotions and expressions. In this section we describe environments that could benefit from our methodology. As an example, the STHENOS project [18] already focussed on the development of a methodology and an affective computing system for the recognition of physiological states and biological activities in assistive environments. Kanjo et al. [12] provide a good introduction to, and review of, the different approaches and modalities for emotion recognition in pervasive environments.

Another scenario in the area of cyber physical systems [15] that is event aware and presupposes a user's reaction is a car-driver assistance system: After a potential harmful external event occurs, the choice between waiting for an appropriate reaction from the driver or initiating an automatic response is crucial. A shift in the driver's facial expressions is one indication to wait in the first case.

Cockpit-based scenarios like the NAVIEYES system provide a lightweight architecture for a driver assistance system [22], that could benefit from facial-expression-shift detection as an additional input source to improve detection of driver's intentions. Another example is McCall et al.'s "Driver Behavior and Situation Aware Brake Assistance for Intelligent Vehicles" which adapted the system's reaction based on situational severity and driver attentiveness and intent by using a camera pointed at the driver's head [19]. Doshi et al. provided an overview on systems for driver behavior prediction and intent inference [8].

Korn et al. [14] published their work reagarding gamification in work environments, which applies facial expression analysis with the FER algorithm SHORE from Fraunhofer IIS and a semi-automatic (Wizard of Oz) approach. This is a similar method as in our previous setup.

2.3 Emotional Models and Expressions

Calvo et al. [7] list six main perspectives for understanding emotions: emotions as expressions, emotions as embodiments, cognitive approaches to emotions, emotions as social constructs, neuroscience approaches with core affect and psychological constructions of emotion.

In this study we focus on the theory of emotions as expressions, which is primarily based on the theory of six basic emotions [9], although the number of expressions detected varies between algorithms.

A common approach for detecting emotional expressions involves generating a feature set of facial landmarks or muscle activity [34]. One approach for discrete quantification makes use of action units (AUs), which are part of the facial action coding system (FACS) by Ekman and Friesen [9] and describe a set of activities based on facial muscles. Coding facial expressions of emotions based on the presence of AUs have often been used in FER algorithms [20, 2, 34].

2.4 Facial Expressions

Facial expressions consist of three different phases: onset, apex and offset (see Fig. 3). All three phases have different durations: while on- and offset are typically short, apex is typically the longest phase. Spontaneous expressions often mix these phases resulting in multiple apexes [13]. Facial expressions can be divided in *normal* and *micro-expressions*, the latter sometimes called *leaking expressions* [9, 32]. Although discussion continues about duration as a criterion of differentiation [32], micro-expressions appear to last less than 0.5 s, while normal expressions typically last longer (often exceeding 1 s) [32].

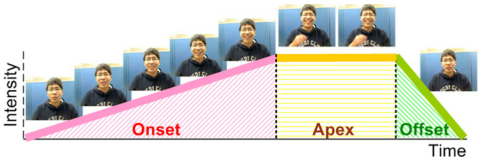


Figure 3: Typical development of a facial expression with onset, apex and offset (from the survey by Chung-Hsien [30], CC BY 3.0).

2.5 FER Algorithms

An overview on the many approaches for detecting facial expressions through image and video processing can be found in the surveys of Zeng et al. [34] and Sariyanidi et al. [27]. State-of-the-art FER algorithms use a pipeline beginning with the crucial first step of finding the face, followed by reducing the data size with filtering [34]. Features are then extracted from the reduced data and machine learning or statistical classification generates the result. Our previous work [3] contains further insights into the nature of FER algorithms.

Algorithms may be trained to detect AUs [20] or facial landmarks [21] as an intermediate step or they may be directly trained for facial expression detection on raw input [29]. Their output is typically an independent probability value between 0.0 and 1.0 for every possible expression (see [3] for details).

Some research has been conducted on the (spatio-) temporal modelling of low-level AUs to exploit their chronological sequence [13] while others [16] have focused on the dynamics of higher level expressions, namely the six basic emotions and a *neutral emotion*.

While much work has been provided on converting video data to FER-output, we found very few studies [1, 5] explaining the output's automatic classification, although it is a necessary step for application integration. We have found no general solution for this post-processing step other than semi-automatic or manual processing.

3. CLASSIFICATION ALGORITHMS

We developed three different algorithms for classifying the FER output data, starting with a primarily mean-based algorithm using a fixed window size. The second algorithm is based on a common approach in signal processing, which uses a matched filter with a fixed scan size and correlation to the data points. The third approach uses Bayesian change-point detection. As our goal was to automate the analysis of our data for the event-based setup, all three approaches were developed for an analysis window around the event.

3.1 Categorising of Data

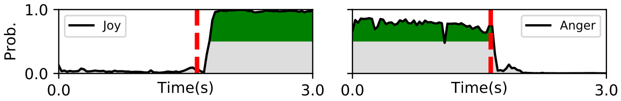


Figure 4: Emotional shift example for two opposite expressions: (a) fast onset of 'joy' and (b) corresponding offset of 'anger'. The red line marks the event position.

The classification of our data is binary-based and coded with two symbols (e.g. "01", see Table 1). The data was subdivided by window size (see Section 4.1) and classified using four different methods: three algorithm-based approaches and manual classification by a single expert (data analyst) acting as a human observer for comparison.

The observer had the same options for classification as those used by the algorithms. The two symbol binary result was used for detecting the *emotional shift* in the facial expression channels (see Fig 4).

Table 1: Types of data classification: "00" denotes that no expression was found, "01" that a rising edge was found, "10" that a falling edge was found, "11" that a stable signal near to 1.0 was found and "??" that the signal was inconclusive.

Category	Example Data
00	____
01	___/‾‾
10	‾‾‾\__
11	‾‾‾‾
??	/‾\/---/

3.2 Peak Detection

All three algorithms use the same method for peak detection that was originally developed by Eli Billauer [4]. We used his standard peak detect version with a look-ahead value of 1 and a delta value of 0.25. After peak detection with a delta of 0.25, we used a value of 0.5 for thresholds between the observed minima and maxima to verify the detection and improve the overall detection rate compared to increasing the peak detection threshold itself. In our previous work [24] we applied the peakutils algorithm by Bergman [25], but preliminary testing revealed that the Billauer algorithm provided slightly better results with this data set.

3.3 Edge-Detection-Based Algorithms

Our proposed edge-detection-based algorithms (CP, PMP) use a common design with a multi-step approach, presented in Figure 5. The main differences between both approaches are the method for processing the data, the threshold for detecting the peaks and edges. These differences are marked in green in Fig. 5. Both methods are explained below.

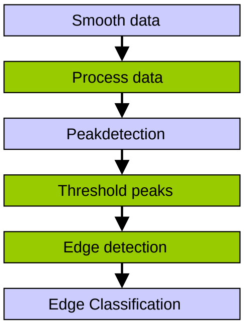


Figure 5: Basic edge-based design: The blue steps are identical in both edge-based algorithms while green steps differ in their processing methods.

3.3.1 *Smoothing of Data*

For the smoothing of the data, we used a modified single-pass moving average filter for each block of (n=4) data points: If the mean is above a threshold of t=0.5, the block value is the maximum value in the block; if the mean is below or equal to the threshold, the block value is set to the minimum instead. This maximises the spread within the block data to improve change detection by the algorithms.

3.3.2 *Processing Data with Changepoint Peak (CP)*

Our changepoint-based design uses Bayesian changepoint detection for identifying the positions of rising and falling edges. We used the changepoint detection method described by Xuan et al. [31], which is based on the work on Bayesian inference for multiple changepoint problems by Fearnhead[10]. We used a constant *prior* of $1/len(data)$ and a *truncate value* of -20 as it produced the best results in preliminary testing on our data set.

3.3.3 *Processing Data with Pattern Matching Peak (PMP)*

Using a simple threshold binary filter is insufficient to process the data, as it still produces a signal requiring additional pattern filtering. We therefore used a reversed approach with pattern filtering and a binary threshold instead.

This algorithm utilises a matched filter [11] based on 1D cross-correlation. We initially used a filter length of $l = 16$ resulting in a complete length of $cl = 2 * l = 32$. This initial filter length was chosen because with a frame-rate of 30 fps it is close to the common minimal length of normal facial expressions (1 s)[32]. We then compared it with smaller (l=8) and bigger lengths (l=24) denominating the algorithm's variants: PMP8, PMP16 and PMP24.

3.3.4 *Edge Detection*

The peak detection process uses a delta value (threshold) that is lower than the actual threshold as described in section 3.2. The edge detection for PMP relies on cross-correlation, which generates separate curves for rising and falling edges.

For CP detection, falling and rising are distinguished by a rating of the two data points before and after the actual edge.

3.3.5 *Edge Classification*

For edge-based methods, edge classification is calculated using Table 2. This table also includes the case of smaller (double) falling or rising edges, if they meet the corresponding condition.

Table 2: Edge classification for CP and PMP algorithms for different number of found edges with corresponding constrain conditions. The conditions ensure the correct order of rising and falling edges.

# Raising edges	# Falling edges	Condition	Result
1	0		01
0	1		10
2	0	rise[0] < rise[1]	01
0	2	fall[0] < fall[1]	10
2	2	rise[0] < rise[1] < fall[0] < fall[1]	10
1	2	rise[0] < fall[0] and fall[0] < fall[1]	10
0	0	mean (left) > 0.5 and mean (right) > 0.5	11
0	0	mean (left) < 0.5 and mean (right) < 0.5	00

3.4 Fixed-Window Mean Bisection (FWMB)

Our binary search based algorithm halves the window around the event position. If no results are found in this iteration, three further subdivision steps are performed. In order to identify possible edges at each depth level, the means of both sides of the window are compared. If the difference between the mean of the left and the right sections is greater than a threshold of 0.5, a direct classification is returned. On each subdivision window, a single peak detection is applied.

3.5 Example Output

Fig. 6 depicts output for all three algorithms as an example. All three algorithms classify this data as "01". Fig. 6 also depicts the main difference between the algorithms: the dependency on window size for detecting edges. FWMB always uses the middle of the window, while PMP and CP are more flexible, as they rely on edge detection rather than a fixed window size.

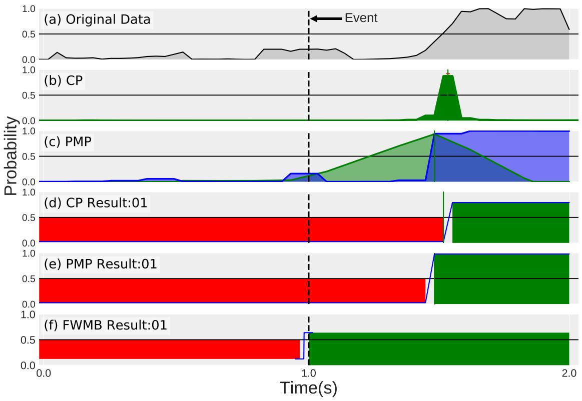


Figure 6: Example output for "01" classification of 'joy': (a) depicts the original data, (b) displays the output from the changepoint detection with peak and raising edge detection, (c) depicts the output from pattern filter and peak and edge detection (green) with smoothed signal (blue) and (d, e, f) provides the results for all three algorithms with mean values in green for values > 0.5 and red if value < 0.5.

4. EVALUATION OF ALGORITHMS

In order to evaluate the three algorithms, we subdivided the data of 2, 4 and 8 s length for a specific facial expression at the event-position: half the window size before and after the event. Every slice of the data was then classified using one of the three developed algorithms and the classification was compared to that of the human observer.

4.1 Experimental Data for Evaluation

Data was collected during experiments in different game levels of the EmotionBike project [24]. In this work, we focussed on two game events: the falling event (fa) from the challenge level (see Fig. 7) and the jump-scare event (js) in the night level (see Fig. 8) resulting in a data set of 92 events and 3,312 subdivided sequences (see Table 3). In general, our exergame involved three types of events:

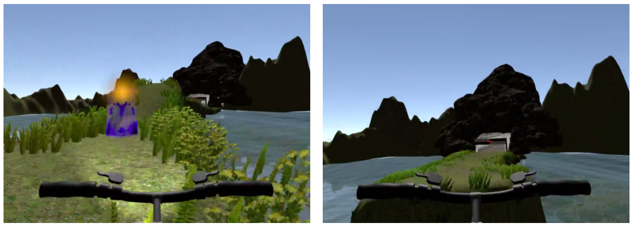
1. **Sudden events**: Users are given no warning when this event will occur, resulting in a small window for detecting facial reactions. The jump-scare event is an example of this type.

2. **Fuzzy events**: Users can estimate the occurrence of this predictable event, making the actual window size larger. The falling event is in this category.

3. **Continuous events**: As the event is constantly present in time, no event time can be calculated. We therefore ignored this type in this study.

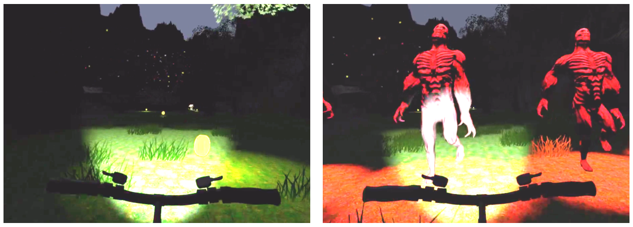
Table 3: Observer-based classification detailing the quantity of each expression for the window sizes of 2, 4 and 8 seconds resulting in a number of 3,312 sequences.

Event	Num of events	Num of expressions	Category	Window		
				2s	4s	8s
fuzzy/task-fail/ falling	81	972	0	535	444	375
			1	167	251	263
			10	196	218	276
			11	65	17	10
			??	9	42	48
sudden/surprise/ Jump-scare	11	132	0	73	60	52
			1	22	27	36
			10	16	36	30
			11	21	7	4
			??	0	2	10



(a) (b)

Figure 7: Challenge/task-fail event: The participant has to bridge a large gap using a ski jump to complete the level. (a) depicts the view before jump, while (b) depicts the view during jump. Task-fail means that the participant drifts to the side while jumping or falls off beforehand and does not reach the ski jump at all.



(a) (b)

Figure 8: Jump-scare/surprise event: The participant rides through a dark forest (a) when suddenly zombies appear as a surprise event without prior warning (b).

We used the Emotient FER algorithm provided as part of the iMotions platform[1] for classifying the video data recorded at 30fps as the algorithm generated good results in our previous benchmarking [3]. We used all 12 provided expression channels, seven basic emotions (joy, anger, surprise, fear, disgust, sadness and contempt) and additionally included: confusion, frustration, neutral emotion, positive emotion and negative emotion.

[1] www.imotions.com

5. RESULTS OF EVALUATION

5.1 Classification Results Table

Table 4 presents the results of the classification for every event, window size and algorithm. Each result is also compared with the corresponding observer classification and has an associated success rate. With the exception of inconclusive data (”??”) a mean is calculated over the other four classifications. An overall mean (all3) is then calculated from the mean values for every window size indicating the best overall success rate for the algorithm and event.

The CP-based algorithm produced the best overall results especially in cases where the total processing frame was 4 s or less. In longer windows, the classification of ”11” often failed when there were small negative spikes often ignored by the human observer.

Table 4: Results with window sizes of 2s, 4s, 8s and a mean over all three window sizes for fuzzy (falling) and sudden (jump-scare) events. This is a condensed table: for the PMP algorithm, only the best results (PMP16) are included. Overall, the CP-based algorithm generated the best result.

Event	Classification	Algorithm	Window			Mean (all3)
			2s	4s	8s	
fuzzy (fa)	00	CP	0.98	0.98	0.90	0.95
fuzzy (fa)	01	CP	0.71	0.76	0.59	0.69
fuzzy (fa)	10	CP	0.79	0.85	0.69	0.78
fuzzy (fa)	11	CP	0.91	0.82	0.30	0.68
fuzzy (fa)	mean	CP	0.85	0.85	0.62	0.77
fuzzy (fa)	00	PMP16	1.00	1.00	0.99	1.00
fuzzy (fa)	01	PMP16	0.51	0.48	0.47	0.49
fuzzy (fa)	10	PMP16	0.57	0.61	0.51	0.56
fuzzy (fa)	11	PMP16	0.98	0.94	0.90	0.94
fuzzy (fa)	mean	PMP16	0.77	0.76	0.72	0.75
fuzzy (fa)	00	FWMB	0.97	0.97	0.92	0.95
fuzzy (fa)	01	FWMB	0.55	0.59	0.59	0.58
fuzzy (fa)	10	FWMB	0.54	0.58	0.57	0.56
fuzzy (fa)	11	FWMB	0.86	0.94	1.00	0.93
fuzzy (fa)	mean	FWMB	0.73	0.77	0.77	0.76
sudden (js)	00	CP	0.99	1.00	0.88	0.96
sudden (js)	01	CP	0.64	0.81	0.64	0.70
sudden (js)	10	CP	0.88	0.97	0.83	0.89
sudden (js)	11	CP	0.86	1.00	0.75	0.87
sudden (js)	mean	CP	0.84	0.95	0.78	0.85
sudden (js)	00	PMP16	1.00	1.00	1.00	1.00
sudden (js)	01	PMP16	0.45	0.67	0.61	0.58
sudden (js)	10	PMP16	0.25	0.36	0.60	0.40
sudden (js)	11	PMP16	0.90	1.00	1.00	0.97
sudden (js)	mean	PMP16	0.65	0.76	0.80	0.74
sudden (js)	00	FWMB	0.90	0.88	0.81	0.86
sudden (js)	01	FWMB	0.73	0.70	0.61	0.68
sudden (js)	10	FWMB	0.44	0.47	0.53	0.48
sudden (js)	11	FWMB	0.90	1.00	1.00	0.97
sudden (js)	mean	FWMB	0.74	0.76	0.74	0.75

5.2 Reliability

Krippendorff’s alpha is a common method for testing the inter-rater reliability [33]. Normally, Krippendorff’s alpha is used to estimate the reliability of a complete group of observers, but it can also be used to compare subgroups [33]. For our purpose, we compared the output of each algorithm with the observer’s result as depicted in Fig. 9.

In all cases, the CP-based algorithms had the highest values. We used Krippendorff’s alpha as an additional criterion for assessing the agreement with the human observer.

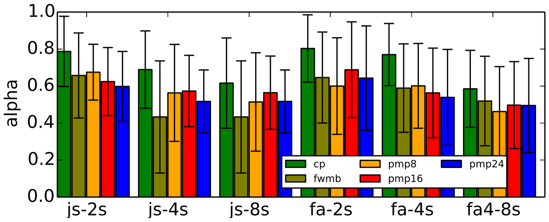


Figure 9: Figure depicting results of Krippendorff’s alpha for all algorithms compared with the observer rating. Results for participant videos were calculated separately for each event (jump-scare=js and falling=fa) and window (2s, 4s, 8s) resulting in an overall mean and standard deviation (SD). The CP-based algorithm produced the best results in case of appropriate windows sizes (2, 4s). Too large window sizes (8 s) degraded the performance.

5.3 Example Confusion Matrices of Classification

Table 5 displays examples of confusion matrices for good (blue), best (green) and worse(red) case classification results of the CP-based algorithm.

Table 5: Confusion matrices for the CP-based algorithm for a 4s sudden event (a) as the best result (overall mean of 0.95 without ”??”-classifications) and (b) a 8s fuzzy event as lowest result (with a mean of 0.55 and with the presence of inconclusive ”??” classifications). Increased results in the ”??” category indicate the analysis window was too large.

(a)

CP sudden (js)	00	01	10	11	???
00	1.00	0.00	0.00	0.00	0.00
01	0.07	0.81	0.11	0.00	0.00
10	0.00	0.03	0.97	0.00	0.00
11	0.00	0.00	0.00	1.00	0.00
???	0.00	0.50	0.50	0.00	0.00

(b)

CP fuzzy (fa)	00	01	10	11	???
00	0.90	0.03	0.06	0.00	0.00
01	0.06	0.59	0.21	0.00	0.15
10	0.01	0.11	0.69	0.00	0.19
11	0.00	0.10	0.50	0.30	0.10
???	0.06	0.21	0.46	0.00	0.27

5.4 Shift Detection Results

Table 6: Results of shift detection. The best match with the observer-based classification is marked in green. The last column contains the success rate of the maximum result compared to the observer’s classification.

Event type	Window	Observer	CP	PMP8	PMP16	PMP24	FWMB
fuzzy (fa)	2s	58	52	54	37	36	35
fuzzy (fa)	4s	68	62	47	50	51	50
fuzzy (fa)	8s	72	59	52	48	51	62
sudden (js)	2s	5	5	3	2	2	3
sudden (js)	4s	8	8	6	5	4	6
sudden (js)	8s	9	8	6	8	8	7

The overall outcome of the shift detection is contained in Table 6. In nearly 92% of cases, the CP-based algorithms matched with the observer classification, although the CP-based algorithm results were lower than PMP-results in one scenario.

We also calculated the Krippendorff's alpha values for comparison between the observer and the different algorithms (see Fig. 10) which further supported the conclusion, that the CP-based algorithms generated the best overall results.

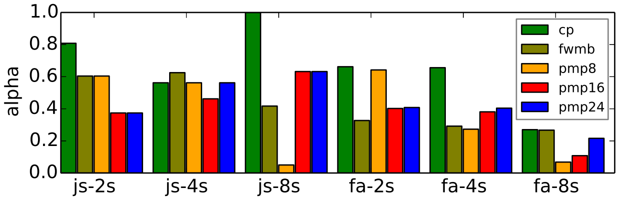


Figure 10: Results of Krippendorff's alpha for all shift detection by algorithms compared to a human observer. No SD was calculated due to the limited number of data points (11 for jump-scare events).

5.5 Performance and Limitations

All three algorithms were capable of processing event windows in soft real time (processing time < 1s), although the CP-based method is normally used for offline detection and had the longest processing time of all three algorithms. For maximum performance, the complete window of data needs to be present.

All three approaches are based on the assumption that changes in facial expressions occur rapidly; and were therefore unable to detect gentle transitions. This was no problem in our case, since our externally provoked events generally occur rapidly.

The integrated CP algorithm has a complexity of $O(n^2)$, which must be considered when increasing the window size or frame-rate of data.

6. CONCLUSION

In this paper, we provide automatic solutions for the classification of facial expression recognition outputs for practical applications. We developed post-processing methods that observe both single-channel and multi-channel shifts as candidate indicators, which can be utilized as a more robust event-response detection. This work addresses one standing research challenge for a fully automatic and unsupervised classification of facial expression reactions tailored for specific applications. While our automatic classification is an important step, we still find it challenging to handle inter- and within-subject variations of responses in a generic way.

Of our three approaches, the changepoint-based classification performed best and it was closest to our human observer results and to human perception of the curves, as demonstrated by the best overall classification performance and the highest values for Krippendorff's alpha. Hereby it is important to avoid too wide analysis windows by pretesting, as these degrade performance.

The window size effect is illustrated in Table 4. The large windows-size effect may be intuitively explained with our study, where the classification of windows with an overall length of 8 s was challenging to score, even for the human observer. In this case, the number of inconclusive categorizations increased significantly (especially for the sudden event; see Table 5), suggesting that this window is too wide.

Table 7 summarizes our overall findings for the three different algorithms in terms of complexity, real-time capability and accuracy.

Table 7: Automatic classification algorithm comparison.

Algorithm	Window	Complexity	Runtime	Accuracy
CP	fixed minimal size	High	High	Good
PMP	fixed patter size	Medium	Medium	Medium
FWMB	fixed window	Low	Low	Medium

Our results further suggest that an automatic processing of shift events shows considerable promise as an additional tool to cope with subject variations. Especially the changepoint-based algorithm produced the best results for the detection of emotional shift with a 92% compliance compared to the human observer-based classification.

Our application of CP and PMP classification provides a starting point for further investigations in short micro-expressions and event-based segmentation techniques without fixed window sizes.

Acknowledgments

This work was funded by the Faculty TI of the Hamburg University of Applied Sciences.We express our gratitude to the EmotionBike team for their technical support.

In this study we applied the peakdetect Python package[2] by Sixten Bergman and Bayesian changepoint detection implemented by Johannes Kulick[3]. All product names are the property of their respective owners.

7. REFERENCES

[1] I. Bacivarov and P. M. Corcoran. Facial expression modeling using component aam models. In *2009 International IEEE Consumer Electronics Society's Games Innovations Conference*, pages 1–16, Aug 2009.

[2] M. Bartlett, G. Littlewort, T. Wu, and J. Movellan. Computer expression recognition toolbox. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–2, Sept 2008.

[3] A. Bernin, L. Müller, S. Ghose, K. von Luck, C. Grecos, Q. Wang, and F. Vogt. Towards more robust automatic facial expression recognition in smart environments. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '17, pages 37–44, New York, NY, USA, 2017. ACM.

[4] E. Billauer. peakdet: Peak detection using matlab, Apr. 2005. http://www.billauer.co.il/peakdet.html.

[5] P. M. Blom, S. Bakkes, C. T. Tan, S. Whiteson, D. Roijers, R. Valenti, and T. Gevers. Towards personalised gaming via facial expression recognition. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'14, pages 30–36. AAAI Press, 2014.

[6] J. Broekens, T. Bosse, and S. C. Marsella. Challenges in computational modeling of affective processes. *IEEE Trans. Affect. Comput.*, 4(3):242–245, July 2013.

[2] https://gist.github.com/sixtenbe/1178136

[3] https://github.com/hildensia/bayesian_changepoint_detection

[7] R. A. Calvo and S. D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, Jan 2010.

[8] A. Doshi and M. M. Trivedi. Tactical driver behavior prediction and intent inference: A review. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011.

[9] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, 1978.

[10] P. Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16(2):203–213, Jun 2006.

[11] E. T. Jaynes. *Probability theory: The logic of science.* Cambridge University Press, Campbride, 2003.

[12] E. Kanjo, L. Al-Husain, and A. Chamberlain. Emotions in context: examining pervasive affective sensing systems, applications, and analyses. *Personal and Ubiquitous Computing*, 19(7):1197–1212, 2015.

[13] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1940–1954, Nov 2010.

[14] O. Korn, S. Boffo, and A. Schmidt. The effect of gamification on emotions - the potential of facial recognition in work environments. In M. Kurosu, editor, *Human-Computer Interaction: Design and Evaluation*, pages 489–499, Cham, 2015. Springer International Publishing.

[15] E. Lee. *Cyber Physical Systems: Design Challenges*, pages 363–369. 06 2008.

[16] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615 – 625, 2006. Face Processing in Video Sequences.

[17] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett. The computer expression recognition toolbox (cert). In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 298–305. IEEE, 2011.

[18] I. Maglogiannis. Human centered computing for the development of assistive environments: The sthenos project. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '14, pages 29:1–29:7, New York, NY, USA, 2014. ACM.

[19] J. C. McCall and M. M. Trivedi. Driver behavior and situation aware brake assistance for intelligent vehicles. *Proceedings of the IEEE*, 95(2):374–387, Feb 2007.

[20] D. McDuff, A. N. Mahmoud, M. Mavadati, M. Amr, J. Turcot, and R. E. Kaliouby. AFFDEX SDK: A cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 2016*, pages 3723–3726, 2016.

[21] P. Michel and R. El Kaliouby. Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, pages 258–264, New York, NY, USA, 2003. ACM.

[22] D. Mihai, G. Florin, and M. Gheorghe. Using dual camera smartphones as advanced driver assistance systems: Navieyes system architecture. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '15, pages 23:1–23:8, New York, NY, USA, 2015. ACM.

[23] L. Müller, A. Bernin, C. Grecos, Q. Wang, K. von Luck, and F. Vogt. Physiological data analysis for an emotional provoking exergame. In *Proceedings of the IEEE Symposium for Computational Intelligence.* IEEE, Athens, Greece, 2016.

[24] L. Müller, S. Zagaria, A. Bernin, A. Amira, N. Ramzan, C. Grecos, and F. Vogt. Emotionbike: a study of provoking emotions in cycling exergames. In *Entertainment Computing-ICEC 2015*, pages 155–168. Springer, 2015.

[25] L. H. Negri and C. Vestri. lucashn/peakutils: v1.1.0, Sept. 2017.

[26] R. W. Picard. Affective computing: challenges. *International Journal of Human-Computer Studies*, 59(1):55 – 64, 2003. Applications of Affective Computing in Human-Computer Interaction.

[27] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, June 2015.

[28] K. R. Scherer, T. Banziger, and E. Roesch. *Outlook: Integration and future perspectives for affective computing.* 2010.

[29] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer. The first facial expression recognition and analysis challenge. In *Automatic Face Gesture Recognition and Workshops (FG 2011)*, pages 921–926, March 2011.

[30] C.-H. Wu, J.-C. Lin, and W.-L. Wei. Survey on audiovisual emotion recognition: databases, features, and data fusion strategies. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.

[31] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1055–1062, New York, NY, USA, 2007. ACM.

[32] W.-J. Yan, Q. Wu, J. Liang, Y.-H. Chen, and X. Fu. How fast are the leaked facial expressions: The duration of micro-expressions. *Journal of Nonverbal Behavior*, 37(4):217–230, Dec 2013.

[33] G. N. Yannakakis and H. P. Martinez. Grounding truth via ordinal annotation. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 574–580, Sept 2015.

[34] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, Jan 2009.