Automatic Detection of Pain from Spontaneous Facial Expressions

Fatma Meawad
University of Glasgow, Singapore
Singapore
fatma.meawad@glasgow.ac.uk

Su-Yin Yang
Tan Tock Seng Hospital
Singapore
su_yin_yang@ttsh.com.sg

Fong Ling Loy
Tan Tock Seng Hospital
Singapore
fong_ling_loy@ttsh.com.sg

ABSTRACT

This paper presents a new approach for detecting pain in sequences of spontaneous facial expressions. The motivation for this work is to accompany mobile-based self-management of chronic pain as a virtual sensor for tracking patients' expressions in real-world settings. Operating under such constraints requires a resource efficient approach for processing non-posed facial expressions from unprocessed temporal data. In this work, the facial action units of pain are modeled as sets of distances among related facial landmarks. Using standardized measurements of pain versus no-pain that are specific to each user, changes in the extracted features in relation to pain are detected. The activated features in each frame are combined using an adapted form of the Prkachin and Solomon Pain Intensity scale (PSPI) to detect the presence of pain per frame. Painful features must be activated in N consequent frames (time window) to indicate the presence of pain in a session. The discussed method was tested on 171 video sessions for 19 subjects from the McMaster painful dataset for spontaneous facial expressions. The results show higher precision than coverage in detecting sequences of pain. Our algorithm achieves 94% precision (F-score=0.82) against human observed labels, 74% precision (F-score=0.62) against automatically generated pain intensities and 100% precision (F-score=0.67) against self-reported pain intensities.

CCS CONCEPTS

• **Human-centered computing → Interaction paradigms**; *Empirical studies in HCI*;

KEYWORDS

Affective Computing, Pain Detection, Personal Healthcare Technologies, Facial Expressions, Ambient Intelligence

ACM Reference Format:

Fatma Meawad, Su-Yin Yang, and Fong Ling Loy. 2017. Automatic Detection of Pain from Spontaneous Facial Expressions. In *Proceedings of 19th ACM International Conference on Multimodal Interaction (ICMI'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3136755.3136794

1 INTRODUCTION

Chronic pain is a debilitating condition that significantly impacts on the individuals' physical, social and emotional function. Measured treatment outcomes are reliant on patients' self-reports of pain, medication usage, activity engagement and mood state. Patient self-reports are however subjected to many biases and may not present an objective reference of patient's actual pain and pain-related experiences. Designing applications for self-management of chronic pain that are sensitive to patients' needs, context and affect can address such biases as well as barriers such as treatment costs, accessibility and long wait times. Current pain management applications however have not been designed with integrated features that effectively address the multidimensional nature of pain [7, 15].

Facial expressions have long been acknowledged as a rich universal, non-verbal modality for communicating pain. Capturing spontaneously facial expressions of patients from a mobile device over time appears to have the potential to be a more objective measure of pain information over self-report. Accompanying self-reporting with automated monitoring can provide deeper insights in personalized pain assessment. In addition, meaningful interventions and follow-up dialogues can be designed to support the patients. Utilising the high-resolution front-facing smartphone camera embedded in all smartphones is one promising channel for collecting facial expressions as patients spend their time on their mobile phones.

Pain detection from facial expressions is a complex problem that requires intensive resources. The lack of ground truth data for spontaneous painful expressions in real world settings presents an additional challenge to research in this area. In this paper, we discuss the initial results of a new approach in detecting pain sequences from spontaneous facial expressions. We extend the state of the art in pain detection based on the Facial action coding system (FACS), a widely-used technique that describes facial expressions through a set of well-defined facial muscle movements [3]. In addition, we adapt the Prkachin and Solomon Pain Intensity (PSPI) metric for measuring pain intensity [14] to detect the presence of pain. We discuss the details of our approach and how it is designed to support mobile-based self-management of pain. Testing our algorithm on the UNBC-McMaster painful dataset [11] showed promising results in detecting sequences of pain with very high precision and good coverage. We conclude the paper with an outlook and further analysis of the results in relation to the target scenario of pain tracking in real world settings.

2 THE FACS OF PAIN

Empirical research in studying FACS and painful expressions consistently showed evidence that most information about pain in the

Table 1: Matching point-pair distances from facial landmarks to the core pain FACS. The multiplication by two indicates the presence of left and right values.

Cheek-raise (AU6)	2*cheek-eye, 2*cheek-nose,2*cheek-lip,2* cheek-bottom-lip
Eye-Lid tightening (AU7) Eye Closure (AU43)	2*P (eye-open)
Nose Wrinkling (AU9)	2*nose-cheek, 2*nose-eyes, 2*nose-lips, nose-bottom-lip
Upper Lip raising (AU10)	nose-bottom-lip, 2* bottom-lip-lips, 2*nose-lips, lip-corners

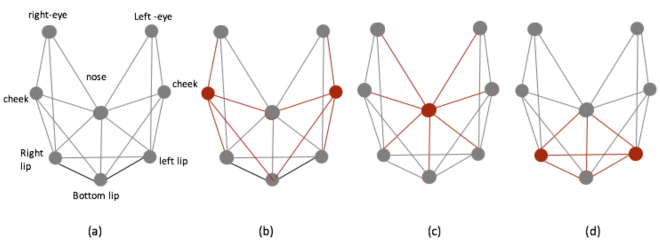

Figure 1: Facial landmarks extracted by Google's Face API (a). The points and distances highlighted in red (b, c and d) show the key points corresponding to pain and their adjacent distances.

human face are conveyed by a few core facial actions: brow lowering (AU4), eye lid tightening (AU7), cheek raising (AU6), upper lip raising (AU10), nose wrinkling (AU9) and eye closure (AU43) [13, 14]. Further studies led Prkachin and Solomon [14] to devise a metric of pain intensity, the PSPI, using the sum of the intensities of the above actions. The PSPI calculations result in a 16-point intensity scale as follows:

$$Painintensity = AU4 + (AU6||AU7) + (AU9||AU10) + AU43 \quad (1)$$

Existing work in detecting pain from facial expressions capitalize on the above findings of the painful FACS. Automatic detection of FACS, however, is a challenge. Trained human experts spend approximately 2 hours to code FACS in one minute of video [9]. The common approaches for automatic detection of FACS combine techniques from computer vision for feature extraction and machine learning for classification [2, 6, 8–10]. Features extracted from the face can be geometric, the shapes and locations of the facial key points, appearances, such as texture, or a mix of both types [17]. Different types of classifiers can be used to recognize expressions on the frame level, such as neural networks and SVM [1], or on the sequence level, such as Recurrent neural networks and Hidden Markov models [4]. Rule-based classifiers were applied on both levels [12]. Putting together all the requirements of the current approaches for AU detection is resource intensive and not suitable for real-time processing on a mobile device. Schiavenato et al. [16] used a light-weight approach in detecting pain for neonates using NFCS, special facial action units defined for infants. The authors showed promising results by relying on parametric statistics in detecting changes in point-pair intensities in video frames.

3 FACIAL PAIN INDICATORS

Building over the findings of Prkachin and Solomon [14], we focus on the local regions of pain in the human face that relate to the four painful action units. We further reduce the complexity of the problem by focusing on clear front views of the face and discarding other poses.

3.1 Feature Extraction

To extract facial landmarks on a mobile device, we use Google's Face API [5]. The Face API can be used for real time detection and tracking of human faces in the mobile camera feed. The API is optimized for mobile use and available for Android and iOS devices. Based on the face orientation, which is given by the API, different landmarks can be extracted from the detected face. In our target scenario, facial expressions are extracted as users spend time using the device. In such case, as users face the front-facing mobile camera, the following landmarks can be extracted: eye centers (left and right), cheeks (left and right), nose base, bottom lip and lip corners (left and right). Moreover, the associated face detector operates at a maximum of 30 frames per second on the mobile device. With such rate, we discard any frames with missing landmarks. The API further characterizes the face with a confidence value of smile and eye open (left and right). However, querying these values require additional computation time. To match the core pain actions to the features extracted by the Face API, we first define the relevant key points: nose, eyes (right and left), cheeks (right and left) and lips (right and left). The probability of eye open (left and right) will be used to indicate eye closure and eye lid squeeze. The Face API does not detect brow lowering; therefore, this specific action unit is not used. Second, the Euclidian distances from each of these points to their adjacent key points (see figure 1) are calculated resulting in 18 unique features. The result is that each pain action unit is described by a set of distances (see Table 2). A total of 18 unique features were identified. Table 2 shows the features and their corresponding action units of pain.

3.2 Feature Analysis

To examine the significance of the extracted features in detecting pain, we used the UNBC-McMaster shoulder pain dataset [11] to perform our analysis. The dataset is publicly available for academic research in pain. It provides 200 video sequences containing spontaneous pain expressions for 25 subjects. The videos are processed into frames, labeled with the PSPI score and FACs. On the session level, the video sequences are labeled by a self-reported pain measure and another one by an observer. Using the Face API on the McMaster dataset generates the coordinates of the eight points of interests in the detected face in each frame (see figure 1). After calculating the Euclidian distances among the key points, we scale the distances as percentages of the detected face's dimensions in each frame.

3.2.1 Identifying pain and no-pain frames. On a frame-level, the only indicator of pain is the PSPI that is calculated based on

Table 2: Facial features ranked based on the mean differences in Z-scores of pain versus no-pain frames.

FEATURE	Z-SCORE MEAN DIFF	CHEEK-RAISE	EYE CLO-SURE	NOSE WRIN-KLE	RAIS-ING LIPS
p(eye-open-right)	-1.55		AU7, AU43		
Lip-left-bottomlip	1.37				AU10
Lip-right-bottomlip	1.25				AU10
cheek-left-bottomlip	-1.05	AU6			
p(eye-open-left)	-1.04		AU7, AU43		
cheek-right-bottomlip	-1.01	AU6			
cheek-lip-left	-0.89	AU6			
cheek-lip-right	-0.79	AU6			
nose-cheek-left	0.71	AU6		AU9	
cheek-eye-left	-0.66	AU6			
cheek-eye-right	-0.61	AU6			
nose-cheek-right	0.60	AU6		AU9	
lip-corners	0.54				AU10
nose-eye-right	-0.38	AU6		AU9	
nose-lip-right	0.34			AU9	
nose-lip-left	0.32			AU9	
nose-eye-left	-0.26			AU9	
nose-lip-bottom	0.06			AU9	AU10

the manually coded AUs. Our close inspection of a sample of the dataset revealed that many frames are missing clear AUs, such as eye closure or eye-lid squeeze. Missing AUs affects the PSPI metric and in multiple frames it scores faces of pain as zero pain. Therefore, to increase the confidence of the collected pain frames, we collect the ones with intensity level of 3 and above. Similarly, for pain-free frames, we collect the ones with no observed AUs to increase the confidence of neutral faces. Six subjects did not satisfy the above requirements; thus, they were dropped from our calculations. In addition, the average number of pain frames dropped from 185 to 74 per subject; while the average number of pain-free frames dropped from 1015 to 878 per subject, with significant variations across subjects.

3.2.2 Standard score analysis. Z-scores were calculated on the scaled features per subject using the means and variances from the no-pain sequences. The process was repeated for 19 subjects in the dataset. Table 2 shows the features ranked based on the mean differences in their Z-scores for pain versus no-pain across all subjects. The rank shows that the top influential features are related to eye closure, cheek raising and lip action. The relationships between the mean Z-score differences for pain and no-pain in our measurements are consistent with common perceptions of painful expressions. The decrease in probability of open eyes (eye closed) is associated with pain while the opposite is true for the increase in the distances between lips' key points (mouth horizontal stretch). The last five features are excluded as their mean score differences do not indicate significant differences for pain and no-pain frames. It is worth noting that most of the discarded features are related to the nose wrinkling action unit of pain (AU9).

4 AUTOMATIC EXTRACTION OF PAIN

Based on our feature analysis results, we selected 11 features that (a) coincide with the core pain action units and (b) show significance in pain frames versus no-pain frames. To activate any of these features in an incoming frame, the corresponding Z-score must be larger than or equal to the absolute of the corresponding Z-score in pain frames. This method requires a calibration step to generate a baseline for each user. To put this requirement in context, users are required to perform a calibration task in their first time of use of the application. Based on this calibration step, a profile is generated for each user consisting of the Z-scores, means and variances of their pain and no-pain expressions.

4.1 Adapting the PSPI

We extend the PSPI formula to detect the presence of pain by substituting the AUs in the formula with the corresponding distance measurements as follows:

$$\begin{aligned} Confidence\_of\_pain = {} & p(eye\_open) + (nose\_cheek) \\ & + (cheek\_eye || cheek\_bottomlip || cheek\_lip) \\ & + (bottomlip\_lip || lipcorners) \end{aligned}$$

Instead of summing the intensities of the AUS, we count the activated features for each AU. We use this as a metric to provide a confidence level for the presence of pain. The presence of any of the cheek point related features (left or right) will add one. Similarly, bottomlip-lip (left or right) or lip-corners will increment one. Any of the eye closure features will add one. To indicate the activation of an AU, at least one feature from its set of features group should be activated. Therefore, the maximum value for this formula is 4 indicating high confidence of pain presence. The p(eye_open) refers to the probability that the left eye or the right eye is open. This value is given by GoogleâĂŹs FACE API.

4.2 Windows of Pain

The captured data from the mobile camera feed in uncontrolled settings does not provide carefully processed sequences of frames as exists in current datasets. Moreover, since our work is based on distances, the results could be highly sensitive to noisy data, which is expected to be frequent in real world settings. Therefore, in addition to dropping any frames with missing landmark data, we detect pain based on its presence in a continuous sequence with a pre-defined size N. If the features are activated in one frame and not the following N-1 frames, then the sequence is discarded. Such approach helps us in avoiding variations in the data caused by intermittent movements, blurry faces or from any other uncertainties in the real world.

Table 4: A confusion matrix of the actual labels created by self reporting, human observers and PSPI versus the detected labels by the proposed algorithm.

	Observed		PSPI		Self-reported	
	Pain	No Pain	Pain	No Pain	Pain	No Pain
Pain	30	11	23	20	23	20
No Pain	2	74	8	120	0	27
	P=94%,R=73%		P=74%, R=53%		P=100%, R=53%	

Table 3: The total number of sequences labeled as pain versus no-pain by the patient, an observer, the maximum PSPI and our algorithm.

	Observed	self-reported	PSPI	our approach
No Pain	78	27	128	141
Pain	93	144	43	30

4.3 Initial Results

In this section, we discuss the results of applying our approach on the UNBC-McMaster dataset. Pain sequences were extracted from 171 video sessions for 19 subjects. The Z-score baseline for each subject was created for the feature activation test that is done per frame. Afterwards, the PSPI formula was applied with highest confidence and pain sequences were extracted using a window size of 10. For the ground truth, we extracted the sequence-level scores given by an observer (0-5 scale) and the ones self-reported by the patient (0-10). Moreover, we extracted the maximum value of PSPI in each video session as a sequence label for it.

Comparing the number of pain/no-pain sequences in each label revealed important observations about the dataset (see Table 3). There is a considerable difference in the number of sequences labeled as pain versus no-pain among the three categories. The bias of self-reporting towards indicating pain is clear; while, the PSPI and our approach show bias towards no-pain. As discussed in section 3.2.1, our definition of pain versus no-pain is based on the frame-level labels, the PSPI, with values bigger than or equal three. Only these frames are used when generating the baselines for the subjects. This explains our approach's bias to no-pain as well.

To accurately measure our algorithm's performance, we calculated the confusion matrices for each label (see Table 4) considering only the sequences that share our definition of pain and no-pain. In other words, if a sequence is labeled with score 4 by an observer, yet the PSPI maximum score is zero, then this frame will not be included in the count. Doing so decreased the total number of pain sequences to 41 for the observer and to 43 for self-reported ones. Based on the calculated confusion matrices, our algorithm achieved 94% precision, compared to human observers (F-score = 0.82), 100% compared to self-reported intensities (F-score= 0.67) and 74% with the maximum PSPI labels (F-score = 0.62).

As shown in table 4, our algorithm missed 20 sequences of pain compared to PSPI. The maximum pain intensity in the un-detected sequences, however, is 6 on an intensity scale of 16 points. Putting this in context, as our algorithm works in the background of users' mobile active time and with a 30-fps rate, the impact of missing such low intensity pain sequences over time is low. Moreover, in this test, we used the highest confidence of our adapted PSPI formula to detect pain. Working with a lower confidence can decrease the number of misses. In all the pain sequences detected by our algorithm, there were no pain-free frames. However, we indicate that there are 8 misses, (see Table 4 for PSPI), since the frame intentisites in those sequences are lower than 3.

5 DISCUSSION

This paper presented a new approach for detecting sequences of pain from video frames. The discussed approach is designed with the requirements of mobile-based self-management of pain in mind. Our approach capitalizes on the latest findings in FACS of pain where pain presence and intensity can be identified from four core facial actions. We used a mobile optimized technique to extract the related key points from a video feed. Significant changes in the distances are subsequently tracked using standardized measurements. The features were activated against a baseline of standardized no-pain measurements that is automatically generated per subject. Testing our approach on a widely-used pain dataset for spontaneous facial expressions showed very high precision and good coverage. The real value of this work, however, requires testing on spontaneous data collected over time in real world settings. Unfortunately, there is no ground truth data with such criteria.

To become successfully immersed in patients' lives, the implementation of this work must be resource efficient, non-obtrusive, yet controllable by users. We implemented a mobile service that tracks the facial expressions of mobile users through the front-facing mobile camera. The service stops tracking if the user is not active or not looking at the device. The service does not require interaction except at installation, as it triggers first time users to perform a 90-seconds calibration task. The frames collected during the calibration task are used as the baseline specific to the user. Moving forward, the mobile service will be used in controlled and uncontrolled longitudinal studies to evaluate its real impact in tracking patients' expressions in relation to self-management of pain. The vision is to use the pain sequence detector to enhance the follow-ups and interventions scenarios in an existing mobile application built by the authors for self-management of chronic pain.

REFERENCES

[1] Marian S Bartlett, Bjorn Braathen, Gwen Littlewort-Ford, John Hershey, Ian Fasel, Tim Marks, Evan Smith, Terrence J Sejnowski, and Javier R Movellan. 2001. *Automatic analysis of spontaneous facial behavior: A final project report.* Technical Report. Technical Report UCSD MPLab TR 2001.08, University of California, San Diego.

[2] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. 1999. Classifying facial actions. *IEEE transactions on pattern analysis and machine intelligence* 21, 10 (1999), 974–989.

[3] Paul Ekman and Wallace V Friesen. 1977. Facial action coding system. (1977).

[4] Irfan A. Essa and Alex Paul Pentland. 1997. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE transactions on pattern analysis and machine intelligence* 19, 7 (1997), 757–763.

[5] Google. 2016. Google Mobile Vision. https://developers.google.com/vision/. (2016). [Online; accessed 16-May-2017].

[6] Zakia Hammal and Jeffrey F Cohn. 2012. Automatic detection of pain intensity. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 47–52.

[7] Chitra Lalloo, Lindsay A Jibb, Jordan Rivera, Arnav Agarwal, and Jennifer N Stinson. 2015. âĂIJThereâĂŹsa pain App for thatâĂİ: Review of patient-targeted smartphone applications for pain management. *The Clinical journal of pain* 31, 6 (2015), 557–563.

[8] James Jenn-Jier Lien, Takeo Kanade, Jeffrey F Cohn, and Ching-Chung Li. 2000. Detection, tracking, and classification of action units in facial expression. *Robotics and Autonomous Systems* 31, 3 (2000), 131–146.

[9] Gwen C Littlewort, Marian Stewart Bartlett, and Kang Lee. 2007. Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain. In *Proceedings of the 9th international conference on Multimodal interfaces*. ACM, 15–21.

[10] Patrick Lucey, Jeffrey F Cohn, Iain Matthews, Simon Lucey, Sridha Sridharan, Jessica Howlett, and Kenneth M Prkachin. 2011. Automatically detecting pain in video through facial action units. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41, 3 (2011), 664–674.

[11] Patrick Lucey, Jeffrey F Cohn, Kenneth M Prkachin, Patricia E Solomon, and Iain Matthews. 2011. Painful data: The UNBC-McMaster shoulder pain expression archive database. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 57–64.

[12] Maja Pantic and Leon JM Rothkrantz. 2000. Expert system for automatic analysis of facial expressions. *Image and Vision Computing* 18, 11 (2000), 881–905.

[13] Kenneth M Prkachin. 1992. The consistency of facial expressions of pain: a comparison across modalities. *Pain* 51, 3 (1992), 297–306.

[14] Kenneth M Prkachin and Patricia E Solomon. 2008. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain* 139, 2 (2008), 267–274.

[15] Benjamin A Rosser and Christopher Eccleston. 2011. Smartphone applications for pain management. *Journal of telemedicine and telecare* 17, 6 (2011), 308–312.

[16] Martin Schiavenato, Jacquie F Byers, Paul Scovanner, James M McMahon, Yinglin Xia, Naiji Lu, and Hua He. 2008. Neonatal pain facial expression: Evaluating the primal face of pain. *Pain* 138, 2 (2008), 460–471.

[17] Ying-Li Tian, Takeo Kanade, and Jeffrey F Cohn. 2005. Facial expression analysis. In *Handbook of face recognition*. Springer, 247–275.