Provisional chapter

Detecting Micro-Expressions in Real Time Using High-Speed Video Sequences

Radu Danescu, Diana Borza and Razvan Itu

Additional information is available at the end of the chapter

Abstract

Micro-expressions (ME) are brief, fast facial movements that occur in high-stake situations when people try to conceal their feelings, as a form of either suppression or repression. They are reliable sources of deceit detection and human behavior understanding. Automatic analysis of micro-expression is challenging because of their short duration (they occur as fast as 1/15–1/25 of a second) and their low movement amplitude. In this study, we report a fast and robust micro-expression detection framework, which analyzes the subtle movement variations that occur around the most prominent facial regions using two absolute frame differences and simple classifier to predict the micro-expression frames. The robustness of the system is increased by further processing the preliminary predictions of the classifier: the appropriate predicted micro-expression intervals are merged together and the intervals that are too short are filtered out.

Keywords: micro-expression spotting, image differences, affective computing, random forest classifier, detection

1. Introduction

Automatic facial expression analysis has been extensively studied in the last decades, as it has applications in various multidisciplinary domains, ranging from behavioral psychology, human-computer interaction, deceit detection, just to name a few. In the last years, a new research field has drawn the attention of computer vision researchers: micro-expression analysis.

Micro-expressions (ME) were discovered by Paul Eckman [1] and his colleagues in the early 1970s while analyzing facial expressions in order to recognize concealed emotions. Eckman defined various facial cues that can be used for deceit detection: micro-expressions, squelched expressions,

and facial asymmetries and various parameters related to the dynamics of the expression. Nowadays, automatic expression and micro-expression analysis have a strong impact on a variety of applications. As an example, in the United States, within the SPOT program [2], airport employees are trained in ME recognition in order to detect the passengers with suspicious behavior. MEs are short facial expressions (with a duration between 1/5 and 1/25 of a second) that usually occur when people try to hide their feelings (either consciously or unconsciously). A micro-expression can be defined by its time evolution, its amplitude, and its symmetry. There are three key moments in the elicitation of a ME: onset (the moment when the ME starts), apex (the moment of maximum amplitude) and offset (the moment when it fades out).

Recently, the automatic analysis of ME has received the attention of researchers in the computer vision field. Besides the difficulties posed by facial expression detection and recognition in general, micro-expressions bring several other challenges. First of all, as MEs are involuntary, data are very hard to gather. However, several ME databases are available [3–5], but they only contain video sequences captured in controlled scenarios. Another difficulty is related to data labeling, as this is a time-consuming and subjective process. As a result, some ME databases [5] classify the expressions only into three categories: positive, negative, and surprise. Finally, MEs are very fast movements and are visible only for a limited number of frames. Therefore, high-speed cameras and accurate motion and tracking algorithms are required in the analysis of ME.

In this chapter, we propose a fast and robust micro-expression detection framework based solely on the movement magnitudes that appear on certain regions of the face. Although numerous works tackled the problem of micro-expression recognition, micro-expression detection has only been addressed recently. However, in real world applications, we argue that ME detection is more valuable than the recognition process.

First of all, the emotion recognition is a complex and fluid problem and psychologists still have not reached a consensus on a taxonomy of emotions and the way they are represented on the face. Eckman proposed a taxonomy with six universal emotions: *anger, disgust, fear, happiness, sadness,* and *surprise*. However, recently, the idea of emotion universality has received much criticism [6]. In practice, more complex emotion classification schemes are used, such as Plutchik's Wheel of Emotions [7] or Parrot's classification scheme [8]. Parrot emotion taxonomy [8] identified more than 100 emotions and classified them into a tree-based structure (primary, secondary, and tertiary emotion levels).

Another problem related to micro-expression recognition is the elicitation of emotion. All the micro-expression data available to this day are captured in (highly) controlled environments: the subjects are asked to watch video sequences with high emotional valence, without moving their head and try to suppress the expression of any emotion. However, using this methodology, the subjects are often impacted by the research technology and pure emotions are not produced, only blended emotions. To solve this problem, Eckman suggests using trained actors in Stanislavski acting technique [1]: in which emotion expression is generated based on actor's conscious thought and past experiences.

Finally, due to micro-expression's short duration and low amplitude, human often fail to perceive them. In fact, in their first study, Haggard and Issacs [9] stated that micro-expressions cannot be perceived with the naked eye.

The proposed algorithm is envisioned to be integrated into a computer-aided emotion analysis system: the detection module determines the frames in the video sequence where the emotion appeared and the psychologist analyzes these frames in order to recognize (a more nuanced) emotion.

The proposed algorithm determines if a ME has occurred at a certain time moment, while the recognition process establishes the type of the micro-expression. For the detection part, we use a sliding window to iterate over the movement variations of the video sequence and we compute the minimum and maximum response for each window position. The resulting feature vector is fed to a classifier in order to determine if a ME occurred at the center of the window. The raw result from the classifier is further processed in order to filter out false positives and to merge responses corresponding to the same ME.

This work has the following structure: in Section 2, the recent advances in the field of ME detection and recognition are presented. The outline of the proposed solution is illustrated in Section 3 and detailed in Section 4. The experimental results are presented and discussed in Section 4. Finally, this work is concluded in Section 5.

2. State of the art

Although automatic ME detection and recognition is not as widely studied as macro-expression analysis, with the recent advances in computer vision, several works addressed this problem. A ME analysis framework usually consists of three main tasks: (1) the selection of the relevant face regions, (2) the extraction of spatiotemporal features, and (3) the detection and recognition of ME using machine learning algorithms.

The first module is related to the selection of the facial areas where the MEs are more likely to occur. The Facial Action Coding System (FACS) [10] is a methodology used to classify facial expressions based on the muscles that produce them and it is used by trained human practitioners. For the automatic ME analysis, the face is usually segmented according to the most prominent facial elements (eyes, mouth corners, and nose) [11–13], or a complex deformable model is used to divide the face into more precise regions [6, 14]. Another approach is to split the face into n equal cells [15, 16].

As MEs are brief facial movements, their analysis requires robust spatiotemporal image descriptors. Various descriptors have been used in the literature: Local Binary Patterns in Three Orthogonal Planes (LBP-TOP) [6], 3D histogram of oriented gradients (HOG) [11, 12], dense optical flow [14], and optical strain [15]. Finally, using the appropriate features, ME can be classified using supervised [6] or non-supervised [11, 12] machine learning algorithms.

Several works perform both ME detection and recognition. In [17], the authors propose a general micro-expression analysis framework that performs both micro-expression detection and recognition. The detection phase does not require any training and exploits frame difference contrast to determine the frames where movement occurred. First, the authors define 36 facial cells based on the position of three facial features (the right eye inner corner, the left eye

inner corner, and the tip of the nose). Two types of features are extracted for the detection process: Histogram of Oriented Optical Flow and LBP histograms; these features are extracted from each facial cell and concatenated into the final feature vector. Finally, the detection module uses histogram differences and thresholding to spot micro-expressions in the high-speed video sequence. The recognition algorithm implies a face alignment preprocessing step. Also, Eulerian motion magnification is used to emphasize the motion magnitude. Next, the classification features are extracted and concatenated from each one of the 36 facial cells: (LBP-TOP, Histogram of Oriented Gradients on Three Orthogonal Planes and Histogram of Image Gradient Orientation on Three Orthogonal Planes). A linear support vector machine classifier is used to recognize the micro-expression type.

In [16], micro-expressions are detected and recognized using optical strain and LBP-TOP motion descriptors. The face region is divided geometrically into 25 rectangular cells, and the feature vector is defined by concatenating the optical strain information and LBP-TOP information from each cell. Finally, a support vector machine classifier is used to both detect and recognize the micro-expressions.

Deep learning and convolutional neural networks in particular have recently received an increasing attention from the scientific literature. Several recent works also tackled the problem of micro-expression detection and recognition from a deep learning perspective. In [18], a convolutional neural network is used to locate 68 features on the subject's face. Based on the position of these landmarks, several regions are defined on the face and the histogram of oriented optical flow (HOOF) is extracted from each facial cell. Finally, the features from all the cells are concatenated and the support vector machines are used to determine the frames in which micro-expressions occurred. In [19], convolutional neural networks and long short-term memory recurrent neural networks (LSTM networks) are used to recognize the micro-expressions. First, a convolutional neural network is trained using the video frames from the beginning of the micro-expression sequence and the onset, apex, and offset frames. The learned features are finally fed to LSTM network to recognize the type of the micro-expression.

As stated earlier, the main problem in micro-expression detection and recognition is the gathering of a representative dataset. **Table 1** describes the micro-expression databases available.

The Polikovsky dataset [13] was captured at a frame rate of 200 fps and involves 10 University students (5 Asian, 4 Caucasian, and 1 Indian). Its main drawback is that the emotions are posed: the students were asked to perform seven basic emotions with low amplitude and go

Dataset	Posed/genuine	Image resolution	FPS	Annotation
Polikovsky [13]	Posed	640 × 480	200	Action units
CASME [3]	Genuine	1280 × 720, 640 × 480	60	Action units 8 emotions
CASME II [4]	Genuine	640 × 480	200	Action units 8 emotions
SMIC [5]	Genuine	640 × 480	100	3 emotions

Table 1. Distribution of the ME types in the CASME-II and SMIC-E databases.

back to the neutral state as fast as possible. Therefore, some difference between these expressions and genuine micro-expressions might occur.

However, some datasets that contain genuine micro-expressions were developed. The following methodology was used to elicit emotions: the users were asked to watch several videos with high emotional valence and try to hide or suppress all their facial expressions that might occur during the experiment. In order to create a high stake situation (as micro-expressions only occur when people have something to lose), some kind of penalty was imposed: if the subjects failed to hide their expression, they would have to fill in a very long and boring questionnaire.

The SMIC [5] database contains 168 micro-expression video sequences labeled with only three emotion classes: positive, negative, and surprise. The dataset was collected using 16 subjects. In addition, 10 subjects were used to capture video sequence at a regular temporal resolution (25 fps) with both visual and near-infrared cameras. For the task of micro-expression detection, a new version of the SMIC dataset was published, which contains longer micro-expression sequences (their average duration is 5.9 s).

The CASME II [4] dataset contains video sequences captured at 200 fps of 35 subjects. In total, 247 micro-expressions were elicited. The database was captured in highly controlled laboratory environments and is labeled with eight emotion classes. The CASME-II dataset also contains some samples (annotated with the "repression" label) that correspond to squelched expressions. Squelched expressions also appear when humans try to conceal their feelings, but there are some major differences between squelched expressions and micro-expressions. First of all, squelched expressions are not complete in terms of temporal parameters: the subject usually becomes aware that expresses an emotion and tries to hide it, by rapidly going to the neutral state or with another emotion (often a smile). Micro-expressions occur involuntary and unconscious, are complete emotions (they have a clear onset, apex, and offset) and their duration is shorter.

The main drawback of all the data available to this day is that all the data are captured in unnatural conditions: the subjects are asked to keep their head fixed and not to make any (macro) facial movements.

3. Solution outline

Figure 1 shows the outline of the proposed solution.

The method analyzes the motion variation that occurs across the high-speed video sequence. Two absolute image differences are computed: the difference between the current frame t and the frame t-ε (that describes the noise variation) and the difference between the current frame and the previous frame at distance $\Delta t/2$ (that describes the motion information).

The main issues that need to be addressed in detecting the micro-expressions are related to their short duration and low amplitude; therefore, this task requires sensible and robust motion descriptors. In **Figure 2**, we depict several frames within some micro-expression sequences.

Ideally, the features used to detect micro-expressions should be based on dense optical flow. However, this descriptor is very hard to compute and extract on the face area: the zone is

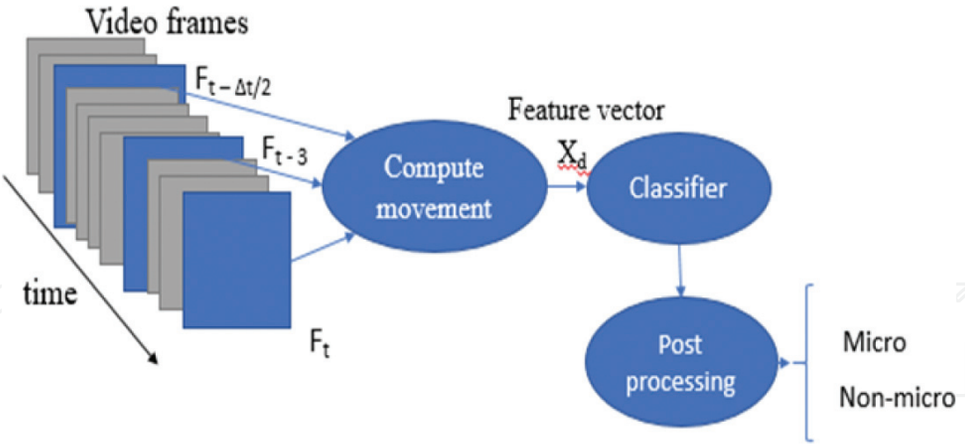

Figure 1. Solution outline.

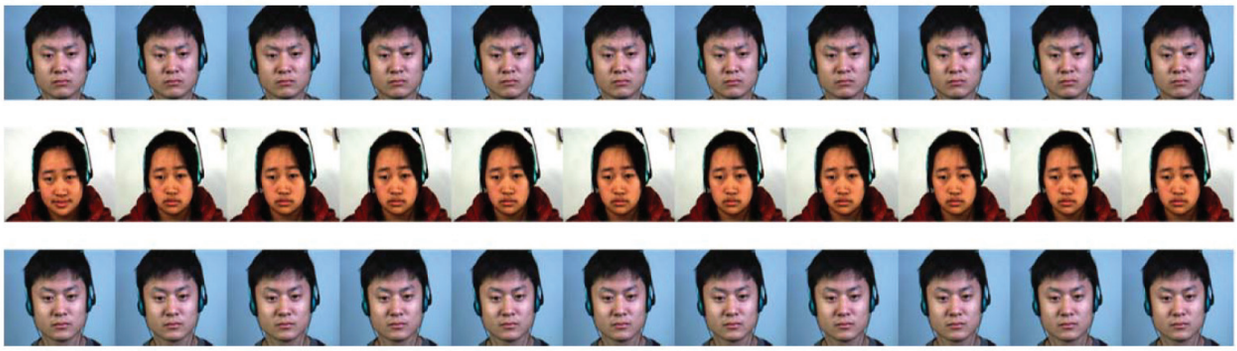

Figure 2. Some samples belonging to micro-expression sequences. First row: an example of a negative ME sequence, second row: an example of a positive ME sequence, third row: an example of a surprise ME sequence. (Raw frames from CASME II database [4] (©Xiaolan Fu)).

mostly homogenous and the micro-expression movement amplitude is too low. We argue that, under these conditions, the dense optical flow is impossible to detect at the pixel level. In addition, dense optical flow computation is slow and requires high computational resources.

The movement magnitude is computed by pixel-wise division of the second difference image by the first difference image. Next, the mean magnitude variation around the most prominent parts of the face (eyebrows, eye corners, mouth corners, and chin) is computed and a classifier is used to determine if a ME occurred at the current frame t. Finally, the response of the classifier is further processed in order to increase the robustness of the solution.

4. Solution description

In this section, a detailed description of each module is presented. First, we describe the regions of interest used to detect the micro-expressions and the computation of the motion detection features. Next, we detail the classification process and the post-processing model used to improve the algorithm's performance.

4.1. Selection of relevant face regions

Our proposed solution analyzes the movement magnitude variation in regions of interest. We defined 10 equally sized regions of the face that correspond to the positions of the muscles that are most used in facial expressions. The selection of the muscles used during a ME was based on the facial action coding system methodology. A first step is to use a general off the shelf facial detector, based on constrained local models [20] to detect 68 facial landmarks. The 10 cells (regions of interest) used in our solution are selected based on the detector results. Therefore, three cells in the upper area correspond to the left frontalis, procerus, and right frontalis muscles (the eyebrows area). Two cells are positioned around the eye corners, corresponding to orbicularis oculi muscles, two cells around the mouth corners and nostrils that overlap the orbicularis oris and zygomatics muscles. The last cell around the chin area overlays the mentalis muscle. The cell dimensions, height and width were chosen heuristically to be half the mouth width. The 10 cells that are analyzed by the ME detection and recognition algorithm are illustrated in **Figure 3**.

4.2. Feature extraction

Our solution relies on a simple method for the estimation of motion variation during a ME. We use Δt to denote the average ME duration (expressed in number of frames) for a given dataset. The facial movements that occur during consecutive frames are very low, as the ME video sequences are captured with high-speed cameras at a high frame rate.

We consider noise as a normalization factor, due to the fact that the movement magnitude for an ME has a very low intensity. Two absolute difference images are computed for each frame t:

1. ΔME (the difference between the frame t and the frame $t - \Delta t/2$).
2. $\Delta\varepsilon$ (the difference between the frame t and the frame $t - \varepsilon$); the resulting images are presented in **Figure 4 (a)** and (b). The $\Delta\varepsilon$ image describes the noise that occurs at frame t.

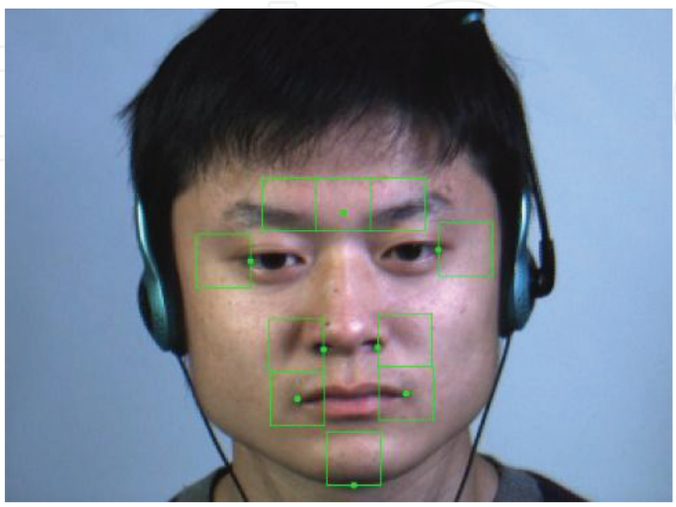

Figure 3. Facial regions of interest. Ten regions of interest are selected around the most prominent facial areas, where the MEs are likely to cause muscle movements.

(a) Δε (b) ΔME (c) M

Figure 4. Frame movement computation. (a) Difference between the current frame and the previous frame at three frames distance. (b) Difference between the current frame and the t the frame t - $\Delta t/2$. (c) Movement magnitude.

The first difference image ΔME describes the movement variation that occurred within the $\Delta t/2$ interval. Due to the fact that there is little to none facial movement within the interval of ε frames in a high-speed capture system, the $\Delta\varepsilon$ image is considered a neutral reference image and it is used as a normalization factor. In the reported results, ε was set to 3; this value was determined through trial-and-error experiments. Therefore, the movement magnitude M (**Figure 4(c)**) at each frame t is computed as:

$$M = \frac{\left|I_t - I_{t-\frac{\Delta t}{2}}\right| + 1}{\left[I_t - I_{t-\varepsilon}\right] + 1}, \tag{1}$$

where I_t represents the frame at index t.

The average value of the M image within the region of interest is computed for each of the 10 face cells (regions of interest). For example, **Figure 5** illustrates the average value of the M image for the middle eyebrow region.

We iterate through the responses for all the cells by using a sliding time window. A feature vector is created using the minimum and maximum values within the time frame and will be further analyzed by a classifier in order to detect if a ME has occurred. For each cell, we compute the

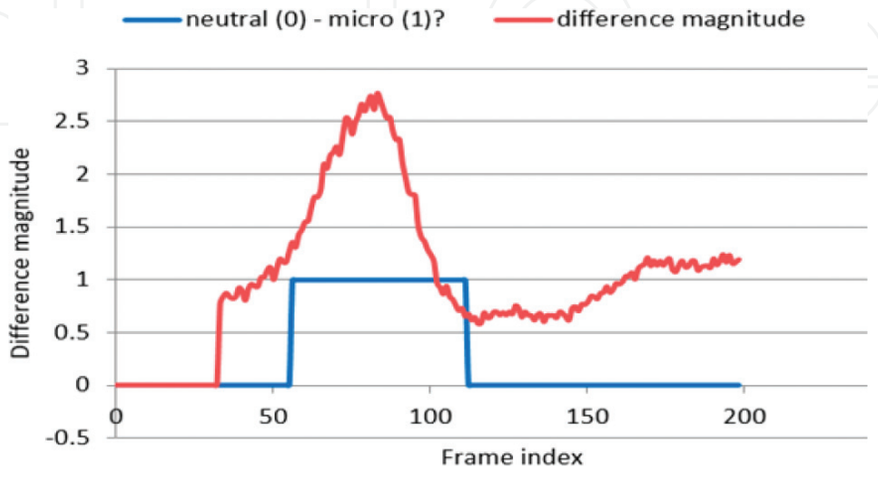

Figure 5. Difference variation of the middle eyebrow face cell. The ground truth labeling of the ME sequence is marked with a blue step, and the difference variation is depicted in gray.

average minimum and maximum value within the sliding window and we concatenate them to the feature vector. The dimensionality of the feature vector is 20 (10 cells × 2 values per cell).

$$feature_t = \|_{c_i \in cell}(\max_{t \in sz}\langle MM_t[c_i]\rangle,\ \ \min_{t \in sz}\langle MM_t[c_i]\rangle), \tag{2}$$

where $\langle MM_t[c_i]\rangle$ represents the average value of the M image within the region of interest c_i, at frame t and || represents the concatenation operator.

We convolved the input frame image with the Laplacian kernel in order to make the algorithm more robust to illumination changes and to eliminate the lighting bias:

$$\mathrm{L} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{3}$$

The image filtered with the Laplacian kernel is presented in **Figure 6**. The results obtained using the raw difference images and the Laplacian filtered difference images are discussed in Section 5.

4.3. Classification

The extracted feature vectors are used as input for a classification algorithm that will determine the state (ME or non-ME) at each frame t. We performed the classification using two classifiers: decision tree and random forest classifier.

Decision trees [21] use structures similar to graphs to determine classification rules, meaning that they are non-parametric supervised learning algorithms. A decision tree's structure contains internal nodes that represent "tests" on an attribute, whereas each edge will represent the outcome of the tests. The leaves in the tree represent the encodings of the class labels, while the classification rules are represented by paths from the root of the tree to each leaf. Decision trees are computationally efficient (the prediction step is logarithmic in the number of data instances used to train the tree), easy to interpret and visualize and require little or no data preprocessing. Their main disadvantage is that the learning algorithm can generate an over-complex tree, meaning that it does not generalize the data well and can usually lead to overfitting.

Figure 6. Laplace filtering.

Random forest classifiers, also known as random decision forest classifiers, [22] are ensemble learning methods for classification, regression, that were designed to cope with the problem of overfitting that occurs in decision trees. These classifiers generate multiple decision trees at training time and the final class label is the mode (the label that appears more often) of the classes of the individual trees. The prediction accuracy is improved by fitting a different number of decision trees on subsets of the dataset and uses averaging to improve the prediction accuracy and to better control overfitting.

4.4. Postprocessing

The preliminary result (R_t) obtained from the classifier is further analyzed in order to filter out false positive and to determine the time frame of the ME (onset, apex, and offset moments). R_t contains the predicted classes (0—non-ME class and 1—ME class) for each frame from the input video sequence. We make the assumption that the preliminary result vector should contain agglomerations of ME class predictions around the apex frame of a ME, and the singular predictions of ME class correspond to false positives. Therefore, we first determine all the contiguous intervals that contain only ME class predictions. The intervals that are too close to each other (their distance is less than $\Delta t/4$) are merged together, and next, all the intervals that are too short (their width is lower than $\Delta t/10$) are considered false positives as filtered out. The remaining intervals are considered ME intervals and their centroid is selected as the apex frame of the ME.

The raw response of the classifiers on an input video sequence and the filtered response of using the proposed algorithm are depicted in **Figures 7** and **8**. The plot is also marked with the ground truth onset, apex, and offset frames.

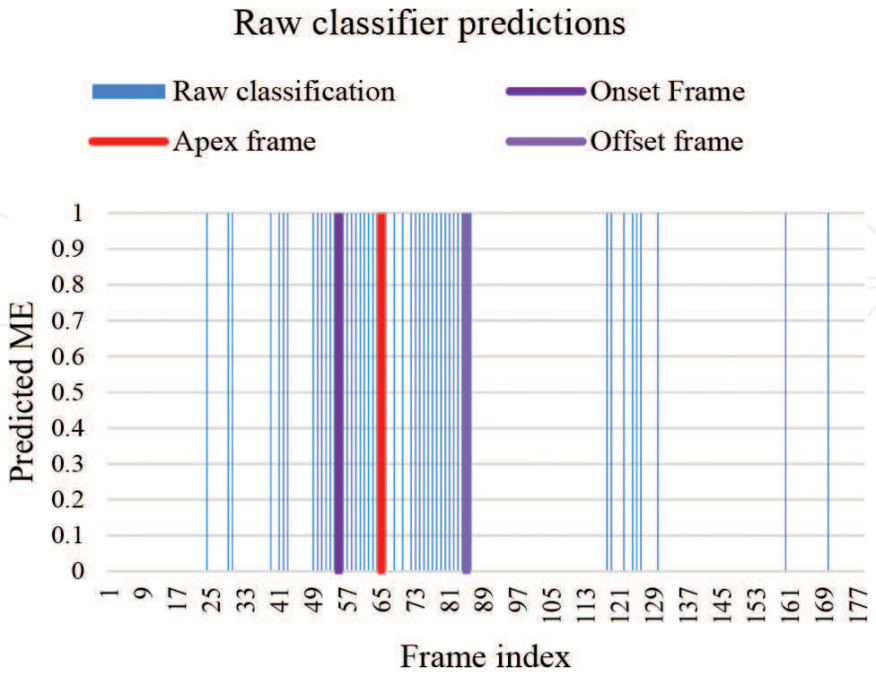

Figure 7. Raw classifier prediction. The predictions are depicted in blue vertical lines; the ground truth onset and the apex and, offset frames are depicted in violet, red, and yellow respectively.

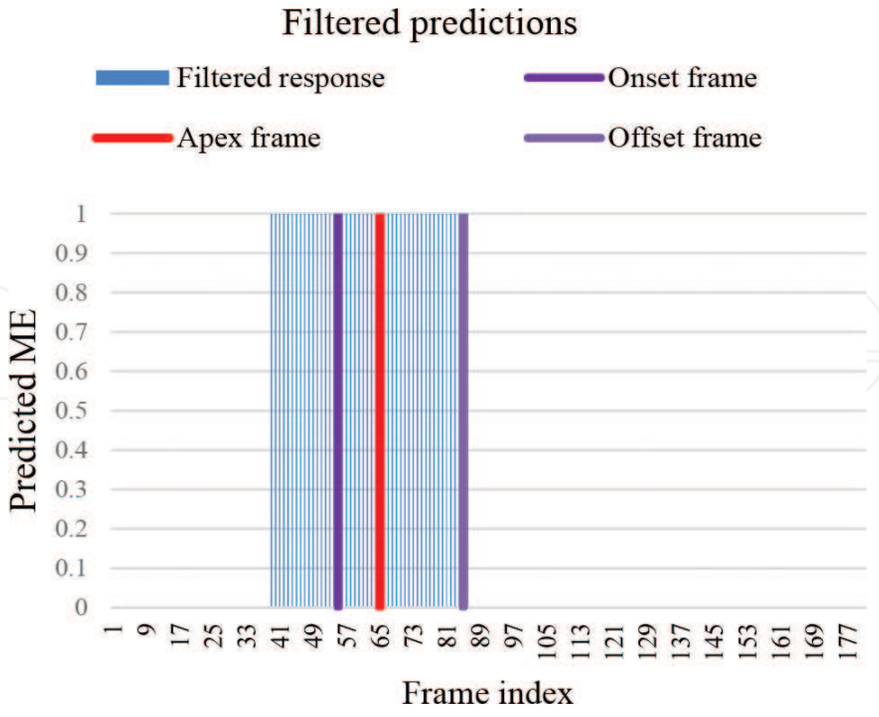

Figure 8. Postprocessing of the classifier result. The retained classifier predictions are depicted in blue vertical lines and the ground truth onset, apex, and offset frames are depicted in violet, red, and yellow respectively.

The classifier response if further post-processed in order to filter out false positives and to merge the positive responses which belong to the same ME. The first step is detecting all the disjunctive ME intervals and then merging together the intervals that are too close to each other. In the last step, the size of each interval is analyzed, and the intervals that are too short are ruled out (Algorithm 1). The middle of each predicted ME interval represent the apex frames.

Algorithm 1: ME detection and postprocessing

Parameters:

minMicroSz: the minimum size in frames of a ME ($\Delta t/4$ in our experiments).

maxDist: the maximum distance between two clusters to be merged ($2\Delta t$ in our experiments).

1: Find the predicted and disjunctive ME intervals: $I = \{(s_0, e_0), (s_1, e_1), \ldots, (s_n, e_n)\}$

2: doMerge ← True.

3: while doMerge do.

4: doMerge ← False

5: for i = 1 to length(I) do

6: $m_1 < - (e_{i-1} - s_{i-1})$

7: $m_2 < - (e_i - s_i)$

8: if $(m_2 - m_1) <$ maxDist then

9: merge(I_i, I_{i-1})

10: doMerge ← True

11: break

Algorithm 1: ME detection and postprocessing

```
12:                 end if
13:         end for
14: end while.
15: for i = 1 to length(I) do.
16:         if (e_i – s_i) < minMicroSz then
17:                 remove(I_i)
18:         end if
19: end for.
```

In the above mentioned algorithm, the predicted ME intervals are described as a list of frame pairs (s_i, e_i) denoting the start and end frames of each interval.

5. Experimental results

The proposed solution was trained and evaluated on the CASME II [5] database. This dataset contains 247 video sequences of spontaneous micro-expressions, captured from 26 participants. The mean age of the participants is 22.03 years, with 1.6 standard deviation. The video sequences were captured by a high-speed camera (200 fps), with a resolution of 640 × 480 pixels. The video sequences are labeled with the onset, apex and offset moments, and with one of following ME types: happiness, disgust, surprise, repression, and tense.

Two types of evaluation strategies are used in the specialized literature: *leave one sample out cross validation* and *leave one subject out cross validation*. The first evaluation technique randomly selects some video sequences for the evaluation, while the latter randomly selects some subjects which were not used in the training process and uses all the samples belonging to the selected subjects for evaluation. Leave one subject out cross validation is more generic, as the classification algorithm hasn't "seen" the subject. For the evaluation part, we used "leave one subject out cross validation" (LOSOCV).

To label the data for detection module, a sliding time window is iterated through the video sequence. If Δt is the average micro-expression duration (67 frames), and t_{apex} is the ME ground truth apex frame, the current frame t is labeled using the following rule:

- If $t \in [0, t_{apex} - \delta \cdot \Delta t]$ or $t \in [t_{apex} + \delta \cdot \Delta t]$, then the frame t is labeled as non-micro-expression frame (neutral frame or macro-expression);
- If $t \in (t_{apex} - \delta \cdot \Delta t, t_{apex} + \delta \cdot \Delta t)$, then frame t is considered a ME frame.

The scale factor δ was heuristically set to 0.25 through trial and error experiments. The main idea regarding this value is that we do not want to label all the frames from the micro-expression interval as micro-expression frames, so we decided that only half the frames from

the ground truth micro-expression interval, centered on the apex frame, to be labeled as micro-expression frames—as there should be a higher movement variation within this region.

Table 2 shows the performance of the algorithm on the CASME II dataset. TPR stands for True Positive Rate, FPR for False Positive Rate, FNR stands for False Negative Rate, and TNR represents the True Negative Rate. The metrics are defined as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{TPR} \tag{5}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

In the abovementioned equations, the following abbreviations are used: *TP*—true positive samples, *FP*—false positive sample, *FN*—false negative sample and *TN*—true negative sample.

The best results are obtained using the Laplace filtering of the input image and a random forest classifier.

Our method is better than recent state-of-the-art methods. In **Table 3**, we present the comparison of the proposed solution with other state of the art works. ACC stands for accuracy, FPR stands for false positive rate and TPR stands for true positive rate.

Feature	Classifier	TPR (%)	FPR (%)	FNR (%)	TNR (%)
Raw pixels	Decision tree	68.18	0.25	31.81	99.74
Raw pixels	Random forest	72.72	0.15	27.27	99.84
Laplacian	Decision tree	76.19	0.06	23.80	99.93
Laplacian	**Random forest**	**86.95**	**0.012**	**13.04**	**99.87**

Table 2. Performance on the CASME 2 dataset.

Method	Features	Performance
[3]	LBP-TOP	ACC: 65.49%*
[5]	LBP-TOP	N/A
[16]	Optical strain, LBP-TOP	ACC: 74.16%*
[17]	Frame differences	TPR*: 70%
Our solution	Frame differences	TPR: 86.95%

Methods marked with an asterisk * were evaluated on SMIC [3] database. To detect the micro-expressions, most of the works were only evaluated on SMIC database. Therefore, the numerical comparison with these methods might not be relevant.

Table 3. Comparison with state-of-the art works.

The execution time of the proposed solution is approximately 9 ms on a fourth generation Intel i7 processor.

6. Conclusions and future work

In this chapter, we presented a fast and robust method for the detection of subtle expressions from high-speed cameras. The method analyzes the movement variations that occur in a given time frame using image differences. Two classifiers were used and evaluated to determine if a ME occurred at a given frame t. In order to ensure the robustness of the algorithm, the raw response of the classifier is further post-processed in order to filter out false positives and to merge the predictions that belong to the same ME zone. The proposed method is fast, robust, and it achieves a high positive rate, while maintaining the false-positive rate low.

As a future work, we plan to gather more data for the training process so that more data variation is present. Till this day, all the micro-expression data are captured in highly controlled environments: artificial lighting conditions, the subjects are not allowed to move their heads freely and must keep a near-frontal pose etc. We plan to gather a different dataset, in which the emotion elicitation technique is quite different (for example, interrogation scenarios) and the users are allowed to act naturally. Of course, under this modified settings, the 3D head pose must be taken into account.

Also, we intend to use motion magnification in order to accentuate the magnitude of the facial movement during a micro-expression and so to increase the algorithm's performance.

Finally, the proposed detection algorithm will be integrated into a full micro-expression analysis framework, which is capable of also recognizing the type of the micro-expression that occurred.

Acknowledgements

This work was supported by the MULTIFACE (Multifocal System for Real Time Tracking of Dynamic Facial and Body Features) of the Romanian National Authority for Scientific Research, CNDI-UEFISCDI, Project code: PN-II-RU-TE-2014-4-1746.

Conflict of interest

The authors declare no conflicts of interest.

Author details

Radu Danescu*, Diana Borza and Razvan Itu

*Address all correspondence to: radu.danescu@cs.utcluj.ro

Technical University of Cluj-Napoca, Cluj-Napoca, Romania

References

[1] Ekman P. Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage. New York: W. W. Norton & Company; 2009

[2] Wikipedia. SPOT (TSA program) [Online], April 28, 2017. Available: https://en.wikipedia.org/wiki/SPOT_(TSA_program)

[3] Rautio H. SMIC—Spontaneous Micro-expression Database, University of Oulu [Online]. April 12, 2017. Available: http://www.cse.oulu.fi/SMICDatabase

[4] Yan W-J, Wu Q, Liu Y-J, Wang S-J, Fu X. CASME database: A dataset of spontaneous micro-expressions collected from neutralized faces. In: FG. IEEE; 2013. pp. 1-7

[5] Zhao G, Liu Y-J, Chen Y-H, Fu X. CASME II: An improved spontaneous micro-expression database and the baseline evaluation. PLoS One. 2014;**9**(1):e86041

[6] Nelson NL, Russell JA. Universality revisited. Emotion Review. 2013;**5**:8-15

[7] Plutchik R. The Nature of Emotions. American Scientist. Archived from the original on July 16, 2001. Retrieved April 14, 2011

[8] Parrott W. Emotions in Social Psychology, Key Readings in Social Psychology. Philadelphia: Psychology Press; 2001

[9] Haggard EA, Isaacs KS. Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy. In: Gottschalk LA, Auerbach AH, editors. Methods of Research in Psychotherapy. Boston, USA: Springer; 1966. pp. 154-165

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

[11] Pfister T, Li X, Zhao G, Pietikainen M. Recognising spontaneous facial micro-expressions. In: 2011 IEEE International Conference on Computer Vision (ICCV); Barcelona; 2011

[12] Polikovsky S, Kameda Y, Ohta Y. Facial micro-expressions recognition using high speed camera and 3D-gradient descriptor. In: 3rd International Conference of Crime Detection and Prevention; 2009

[13] Polikovsky S, Kameda Y, Ohta Y. Facial micro-expression detection in hi-speed video based on facial action coding system (FACS). IEICE Transactions on Information and Systems. 2013;**E96**(1):81-92

[14] Godavarthy S, Goldgof D, Sarkar S, Shreve M. Macro- and micro-expression spotting in long videos using spatio-temporal strain. In: Automatic Face & Gesture Recognition and Workshops (FG 2011); 2011

[15] Liu Y-J, Zhang J-K, Yan W-J, Wang S-J, Zhao G. A main directional mean optical flow feature for spontaneous micro-expression recognition. IEEE Transactions on Affective Computing. 2016;**7**(4):299-310

[16] Liong ST, See J, Phan RC-W, Oh YH, Le Ngo AC, Wong KS, Tan SW. Spontaneous subtle expression detection and recognition based on facial strain. Signal Processing: Image Communication. 2016;**47**:170-182

[17] Li X, Xiaopeng HONG, Moilanen A, Huang X, Pfister T, Zhao G, Pietikainen M. Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods. IEEE Transactions on Affective Computing. 2017. http://ieeexplore.ieee.org/document/7851001/

[18] Li X, Yu J, Zhan S. Spontaneous facial micro-expression detection based on deep learning. In: 2016 IEEE 13th International Conference on Signal Processing (ICSP). IEEE; 2016. pp. 1130-1134

[19] Breuer R, Kimmel R. A deep learning perspective on the origin of facial expressions. arXiv 674, preprint arXiv:1705.01842 2017

[20] Cox M, Nuevo J, Saragih J, Lucey S. CSIRO Face Analysis SDK. AFGR; 2013

[21] Quinlan JR. Induction of decision trees. Machine Learning. 1986;**1**(1):81-106

[22] Ho TK. Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition. Vol. 1. IEEE; 1995. pp. 278-282