Image and Vision Computing 51 (2016) 36–48

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Cascade of Tasks for facial expression analysis ☆

Xiaoyu Ding[a,*], Wen-Sheng Chu[b], Fernando De la Torre[b], Jeffery F. Cohn[b,c], Qiao Wang[a]

[a] School of Information Science and Engineering, Southeast University, Nanjing, China
[b] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, United States
[c] Department of Psychology, University of Pittsburgh, Pittsburgh, PA 15260, United States

ARTICLE INFO

Article history:
Received 21 February 2014
Received in revised form 28 August 2015
Accepted 22 March 2016
Available online 31 March 2016

Keywords:
Automated facial expression analysis
Action unit detection
FACS

ABSTRACT

Automatic facial action unit (AU) detection from video is a long-standing problem in facial expression analysis. Existing work typically poses AU detection as a classification problem between frames or segments of positive and negative examples, and emphasizes the use of different features or classifiers. In this paper, we propose a novel AU event detection method, Cascade of Tasks (CoT), which combines the use of different tasks (*i.e.*, frame-level detection, segment-level detection and transition detection). We train CoT sequentially embracing diversity to ensure robustness and generalization to unseen data. Unlike conventional frame-based metrics that evaluate frames independently, we propose a new event-based metric to evaluate detection performance at the event-level. The event-based metric measures the ratio of correctly detected AU events instead of frames. We show how the CoT method consistently outperforms state-of-the-art approaches in both frame-based and event-based metrics, across four datasets that differ in complexity: CK+, FERA, RU-FACS and GFT.

1. Introduction

Facial expressions convey varied and nuanced meanings. Small variations in the timing and packaging of smiles, for instance, can communicate politeness, enjoyment, embarrassment, or social discomfort [1,2]. To analyze information communicated by facial expressions, Ekman and Friesen proposed the Facial Action Coding System (FACS) [3]. FACS describes facial activity in terms of anatomically based action units, which can occur alone or combine to represent all possible facial expressions. Action units (AUs) have a temporal envelope that minimally include an onset (start) and offset (stop) and may include change in intensity. Researchers have defined 33 to 44 AUs, depending on FACS version [4].

In computer vision, automated AU detection has become an active area of research [6–15] and has been applied to marketing, mental health, instructional technology, and media arts [16–20]. Despite its descriptive power [5], automatic AU detection is challenging: non-frontal pose and moderate to large head motion complicate facial image registration; the temporal scale of facial actions varies considerably; individual differences occur in shape and appearance of facial features; and many facial actions are inherently subtle. Due to the thousands of possible combinations of 30-40 or more AUs, detection typically is decomposed to a binary classification problem on each AU.

Existing AU detection methods broadly fall into one of three types: frame-level detection, segment-level detection, and transition detection. Frame-level detection independently evaluates each video frame for the occurrence of one or more AUs [8,11,13,21–24]. Segment-level detection seeks to detect contiguous occurrences of AU that ideally map onto what manual FACS coders perceive as an event [12,25–27]. Transition detection seeks to detect the onset and offset of each segment, or event [28]. See [29,30] for recent surveys.

Most approaches to AU detection are frame-level detectors, which consider each video frame as independent. Because this assumption ignores the inherent auto-correlation of behavioral data, detection tends to be noisy with classifiers firing on and off in proximal frames. By contrast, human observers do not evaluate video frames individually. Rather, they perceive AU as *events* that have a beginning (onset), an end (offset), and a certain duration. Consequently, manual FACS coding requires significant effort to first perceive an AU event and then identify its precise onset and offset. To identify such events, researchers rely on segment-level detection. Often, it is relatively easy to detect the temporal segment in the middle of an AU event with high intensity or large facial movement, yet the transition points between AU inactivation and activation are more subtle and difficult to detect. We seek to automatically detect

☆ This paper has been recommended for acceptance by Vladimir Pavlovic.

* Corresponding author. Tel.: +1 412 999 8605.
E-mail addresses: leonxd@andrew.cmu.edu (X. Ding), wschu@cmu.edu (W. Chu), ftorre@cs.cmu.edu (F. De la Torre), jeffcohn@pitt.edu (J. Cohn), qiaowang@seu.edu.cn (Q. Wang).

http://dx.doi.org/10.1016/j.imavis.2016.03.008
0262-8856/

AU events, including onsets and offsets, with high fidelity to human perception.

To achieve this goal, we propose a Cascade of Tasks (CoT). CoT detects AU events including their onsets and offsets, by sequentially integrating the three AU detection tasks: frame-level detection, segment-level detection, and detection of onsets and offsets. Fig. 1 illustrates the CoT process. The first task detects AU at the frame-level. The results of this task tend to be noisy, or less reliable, because it fails to exploit the temporal dependencies among proximal frames.

The second task combines the output of the frame-level detection with new segment-level features with a segment-based classifier (see Fig. 1 second row). The segment-level detector gives a rough location of the AU event and reduces the frame-level false positives, but is imprecise in the boundaries (*i.e.*, onset and offset). The third task addresses this problem. By integrating the three tasks, CoT provides a more robust and precise detection of AUs than previous approaches.

Our contributions are two-fold. 1) To the best of our knowledge, CoT is the first approach to integrate multiple *tasks* for AU detection. Most other algorithms for AU detection emphasize different features or a classifier, or combine them with ensemble-type methods to solve a single task. However, our approach combines different tasks.

2) CoT fully recovers AU events instead of isolated AU frames or incorrectly parsed segments.

To evaluate AU detection performance at event-level, we propose a new event-based metric, as opposed to conventional frame-based metrics that evaluate frames independently.

2. Previous work

We broadly categorize AU detection approaches into three types of *task*: frame-level detection, segment-level detection and transition detection. These approaches largely differ on the methods for registration, feature representation, and classifier learning. Here we review recent work on AU detection. Refs. [6,7,29–31] offer more complete surveys.

The first AU detection challenge (FERA) [7] indicates that most approaches, including the winning one, were frame-based. Frame-level methods detect AU occurrences in individual frames by extracting geometric or appearance features to represent each frame, which are then fed into static classifiers (*e.g.*, SVM [8,32] or AdaBoost [11,13]). Geometric features contain information of facial feature shapes, including landmark locations [22,32,33] and geometry of facial components [34]; appearance features capture texture changes of the face, such as wrinkles and furrows, and can be typically represented by Gabor [11,35], LBP [24,36,37] and DAISY/SIFT descriptors [13]. A notable trend in this area is fusing various features/classifiers to generate more accurate and robust results [38,39]. For example, Tariq et al. [40] concatenated image features, including SIFT, Hierarchical Gaussianization and optical flow, as input to a SVM classifier. Later, Tariq et al. [9] used a log sum model to fuse the outputs of classifiers trained separately with different low-level image features.

In their study of multilayer architectures of texture-based image feature descriptors (filters), Wu et al. [21] showed that adding a second layer of nonlinear filters consistently improved performance. This approach represents a special way to fuse feature descriptors. Almaev and Valstar [41] proposed a temporal extension to the multi-layer appearance features (LGBP-TOP). More recently, Jiang et al. [42] proposed a decision-level fusion strategy to combine region-level classifiers. First, domain knowledge regarding FACS AU definition is used to define a face region. Second, a region-specific classifier is trained for each region. Finally, a weighted sum combines outputs of these classifiers.

Segment-level approaches seek to incorporate temporal information of facial action, and to detect AU as a set of contiguous frames. To capture temporal information, dynamic features have been used to measure motions on a face [43,44], such as raising mouth corners. Recent work on exploiting dynamic features includes bag of words [12] and temporal extensions to LBP, LGBP and LPQ [20,23,37,41,45]. Another approach models the AU state change over time using temporal classifiers or models. For example, Chang et al. [25] use hidden conditional random fields to link the AU state with underlying emotions in facial expression sequences. At each time

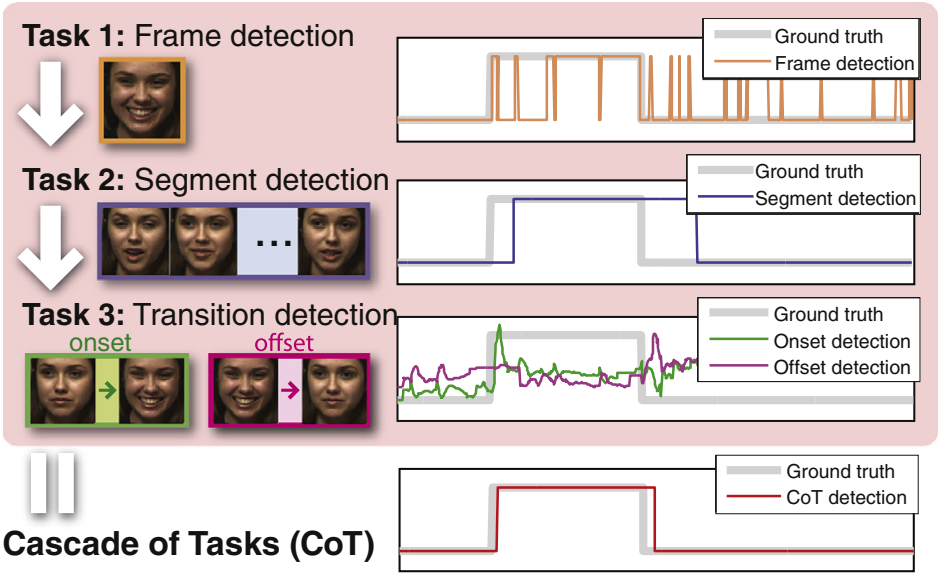

Fig. 1. Detection of AU 12 (smile) from its onset to offset using our proposed CoT method. In the plots to the right above, thick gray lines indicate ground truth and thin lines indicate prediction results. First, CoT detects AU 12 in individual frames (Task 1). Because this step assumes that individual frames are independent, it is prone to error. Next, CoT uses the responses of the frame-level detector and segment-based features to detect a segment for AU 12 (Task 2). Finally, CoT more precisely estimates the onset and offset frames by learning transition detectors (Task 3).

step, inference is made incrementally based on previous inferences. Similarly, Tong et al. [27] use Dynamic Bayesian Networks to model the semantic and temporal relationships between AUs. Simon et al. [12] train segment-based SVM and detect AU segments in image sequences by using dynamic programming.

In general, frame-level detectors are shown to be able to detect subtle AU events because of the sensitivity to each frame. However, they are prone to noise due to lack of temporal consistency. Segment-level methods, on the other hand, better detect AUs than frame-level approaches and better approximate human perception. However, compared with frame-level training data, segment-level training data are scarce. Moreover, AU segments can be temporally complex and are difficult to model. Consequently, segment-level approaches are often less discriminative and cannot detect subtle AU events. In this work, we use frame-level detection output to augment segment-level training data. This results in segment-level detectors with better discriminative power.

Recently, Walecki et al. [46] proposed a Variable-state Latent Conditional Random Field model for segment expression analysis, which can automatically select optimal latent states (nominal or ordinal).

In addition to frame-level and segment-level approaches, an important yet relatively unexplored task is to detect only AU transitions (*i.e.,* onsets and offsets). However, transition detection is challenging due to subtle changes between AU and non-AU frames. To account for these subtle changes, previous approaches have relied on additional information, such as an AU apex location [28] manually labeled by a user, to accurately detect transitions.

Besides the above approaches that detect AU activation (*i.e.*, inactivation and activation), other works analyze AU events by explicitly detecting their constituent temporal phases (*i.e.,* neutral, onset, apex and offset) [45,47–52]. This method provides a more precise and nuanced framework. Pantic and Patras [48] proposed to use geometric based features and temporal rule-based reasoning to recognize temporal phases of AU events. Later, Valstar and Pantic [49,51] proposed a hybrid model of SVM and HMM that combined the SVM's discriminative power and HMM's ability to model time to recognize AU and divide temporal phases.

More recently, motion-based feature and spatial-temporal features [45,50] have been used to detect AU and temporal phases. In addition, Conditional Ordinal Random Field [52] was used to account for ordinal relations between temporal phases of an AU event.

In this work we focus on AU activation detection for its simplicity and wide use. Following Ref. [28], we use "onset" and "offset" to refer to exact frames where transitions between AU inactivation and activation occur.

In other words, onset and offset indicate the beginning onset frame and the ending offset frame. Previous works used the terms to indicate two temporal phases [45,47–52].

Therefore, the transition (onset and offset) detection in CoT can be viewed as one aspect to the overall AU activation problem, while in Refs. [45,47–52] the onset and offset detections are two of four detection goals.

CoT also relates to Multitask Learning (MTL) [53], which defines multiple related tasks and improves learning for one task by using the information contained in the training data of other tasks. However, there are several notable differences. First, in MTL each task can have different objectives, while in CoT all three individual tasks are various aspects of one overall task (*i.e.* AU event detection). Second, the principal goal of MTL is to improve generalization performance by sharing representation among tasks [53], while CoT's goal is to improve the performance of the overall task by combining the three individual tasks roughly at decision-level. Third, MTL trains tasks in parallel, using a shared representation. CoT performs tasks in sequence where a subsequent task benefits from the preceding task by using its output rather than sharing feature representation.

3. Cascade of Tasks (CoT)

Unlike previous AU detection methods that combine features and classifiers for one particular task, Cascade of Tasks (CoT) sequentially integrates three different tasks: 1) **Frame-level detection** for detecting AU presence/absence on bases of information extracted from a single frame; 2) **segment-level detection** for detecting AU segments from contiguous frames; and 3) **transition detection** for recognizing transitions between AU and non-AU frames.

Generally, we first perform frame-level detection and detect an AU event at segment-level. Results are then refined by transition detection.

This order of tasks is based on the assumption that the output format of event-based detection is more desirable than frame-based results by Human Computer Interaction, animation and other applications. Frame-based results tend to be noisy with classifiers firing on and off in proximal frames, whereas event-based results are temporally smoother. Conducting frame-level detection first allows us to use subsequent tasks to enforce temporal smoothness. We also assume that, given the same training expression sequences, training data for frame-level are most ample among the three tasks. In addition, various techniques have successfully represented facial expression images at the frame-level. On the contrary, segment-level training data are relatively scarce and are much more difficult to represent. Results of frame-level detection are then used to augment segment-level training data in both feature representation and training sample weighting. In this way, the proceeding task in the cascade makes each subsequent task easier. Finally, we assume that subtlety of transition in facial expressions complicates transition detection and yields unstable results. Therefore, we only use the transition detection to refine segment-level detection results.

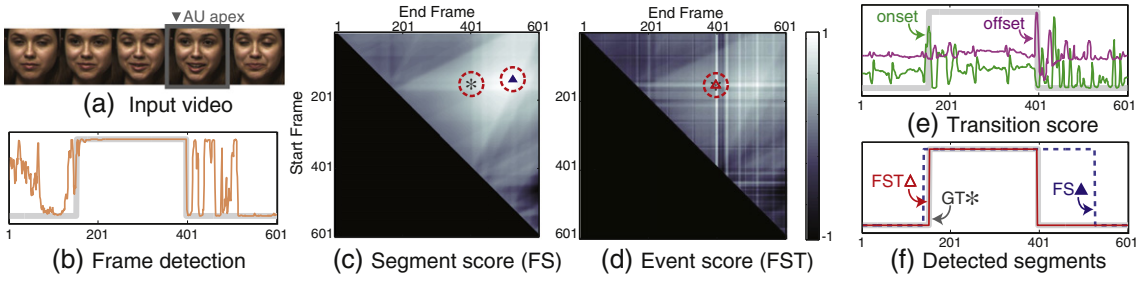

Fig. 2. Example from the RU-FACS dataset [11]: (a) a video of subject 77, (b) frame detection result in thin orange line and ground truth (GT) in thick gray line. (c) A segment score matrix for frame + segment (FS) detection. The higher the score is, the more likely that there is an AU in this segment. (d) Event score matrix for FS + transition (FST) detection. Using the transition score in (e) as a refinement, FST detector (△) fires closer to the GT. (f) Detected segments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1. Frame-level detection

For frame-level detector we use a Support Vector Machine (SVM) trained on appearance features (SIFT descriptors) following Ref. [13]. We train the SVM using a leave-one-subject-out strategy. These frame-level detectors offer reasonable predictions for frames with the AU presence, but often are prone to noise due to the lack of temporal consistency. Fig. 2 (a) and (b) illustrates a frame-based detector on a video of 601 frames that contains an AU 12. Observe that the frame-level detector detects correctly the frames where the AU is present (153 – 398 frames) but has many false positives. While the frame-level detector may contain a large number of isolated false detected frames, they are fast and easy to train. We will use the output of the frame-level detector (f_{frm}) to improve the subsequent task (*i.e.*, segment-level detection).

3.2. Segment-level SVM

To eliminate isolated false detections while preserving the sensitivity of frame-level detectors, we will use the outputs of the frame-level detection in combination with new segment-based features.

3.2.1. Segment-level feature

We divide each segment evenly into three sub-segments, and compute for each sub-segment a temporal bag of words [12] with geometric features [34], as a complement to the appearance features used in the frame-level detector.

The geometric features used here are a set of predefined geometric measurements based on facial landmarks, including distances between facial landmarks and facial components, as well as heights and angles of facial components. Fig. 3 shows several examples of the geometric features. x_1^U and x_2^U are distance between brows and eyes. x_3^U, x_1^L and x_2^L are heights of eye, lip and teeth. x_3^L is angle of mouth corner.

In our experiment, the facial landmarks are defined as in Xiong and De la Torre [54].

Introducing these geometric features promotes diversity among the tasks and hence produces more robust AU detection (as will be shown in Section 5). For each sub-segment, we also incorporate the statistics of the output scores from the frame-level detector f_{frm}. In particular, we include the maximum, minimum, mean and median over the frames that constitute the sub-segment. The final segment-level representation is a concatenation of the histograms of temporal words and frame score statistics from the three sub-segments.

3.2.2. Segment-level detector

Given the segment-level features and the prediction scores from the frame-level detectors, we train the segment-level detector using a weighted margin SVM [55]:

$$\begin{aligned} &\min_{\mathbf{w},\xi_k} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_k v_k\xi_k \qquad (1)\\ &\text{s.t.} \quad \frac{y_k}{v_k}\mathbf{w}^\top\psi(\mathbf{S}[s_k,e_k]) \geq 1-\xi_k,\\ &\xi_k \geq 0, k = 1,\ldots,n, \end{aligned}$$

where n is the number of training segments and $\{\xi_k\}_{k=1}^n$ are the slack variables. $\psi(\mathbf{S}[s_k,e_k])$ denotes a segment-level feature for the *kth* segment, $\mathbf{S}[s_k,e_k]$, starting in the s_k^{th} frame and ending in the e_k^{th} frame. To simplify the notation, we concatenate the segment features $\psi(\mathbf{S}[s_k,e_k])$ with 1 to compensate for the offset. $y_k \in \{-1,1\}$ denote the labels. $\{v_k\}_{k=1}^n$ are confidence weights that give more importance to some segments than others. The higher the v_k the more important the segment will be in the classification process. Recall that in segment-level detection, the positive segments are the manually labeled AU events (of different length and intensity). The negative segments are sampled segments at random locations and temporal scales, and typically outnumber positive segments. For each segment $\mathbf{S}[s_k,e_k]$, we compute the confidence weight as the averaged absolute value of the frame-level detection scores, that is $v_k = \frac{1}{e_k-s_k+1}\sum_{i\in[s_k,e_k]}|f_{\text{frm}}^i|$, where f_{frm}^i is the output of the frame-level detector in *ith* frame. With this definition of confidence weights, we give more importance to the segments that are more likely to contain many frames where the frame-level detector returns higher scores. Please note that the confidence calculated here is frame-level confidence. It is easy to compute and works well in practice. It can give us a hint of how confident we think a segment contains an AU event. However, it is not exactly AU event confidence. Given a segment $\mathbf{S}[s_k,e_k]$, the decision value of segment-level detector is denoted as $f_{\text{seg}}(\mathbf{S}[s_k,e_k]) = \mathbf{w}^\top\psi(\mathbf{S}[s_k,e_k])/v_k$.

Segment-level detectors achieve more robust decision on contiguous frames, but often mis-detect subtle AU events due to insufficient positive events for training, e specially in the onset and offset. Fig. 2 (c) illustrates the score matrix (601×601) of the segment-level detector on a video of 601 frames. Each entry (i,j) of the matrix corresponds to the segment-level score that starts in the *ith* frame and ends in the *jth* frame. The higher the score the more likely the segment contains an AU. In this particular case, the ground truth solution (GT ()) is located at (153,398). However, the

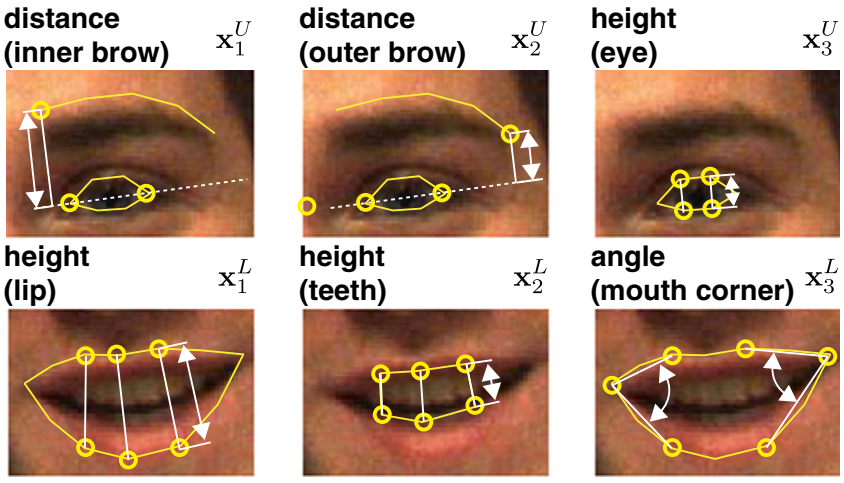

Fig. 3. Examples of geometric feature. x_1^U and x_2^U are distance between brows and eyes. x_3^U, x_1^L and x_2^L are heights of eye, lip and teeth. x_3^L is angle of mouth corner.

segment-based detector (FS detector) (▲) finds the maximum score at (140,526). The segment-level detection (blue dashed line in Fig. 2 (f)) fires in a region that covers both the true AU and the subsequent speech related facial movements. In order to improve the detection around the onset/offset, we will add the transition detection task.

3.3. Transition detection

As discussed above, segment-level detections are often inaccurate in the boundaries (onsets and offsets) of AU events. In this section, we propose a transition detection to refine boundaries of the segments previously detected.

We train two transition detectors, one for onsets and the other for offsets, using linear SVM.

As discussed before, in this work we choose to focus on AU activation detection. Accordingly, onset and offset are referred to exact frames where transitions between AU inactivation and activation occur, instead of temporal phases. To be precise, they are beginning onset frame and ending offset frame. Therefore, the transition detectors are trained to detect the frames in between two contiguous time periods of AU inactivation and activation, instead of AU intensity increase or decrease. For this reason, a multiple-apex AU event, as well as a single-apex AU event, ha ve only two transition frames (*i.e.* beginning and ending), and are treated as one AU event.

We denote the detectors as f_{on} and f_{off} for onset and offset respectively. We collected positive samples by extracting segment-level features in segments centered in the offsets and onsets. We selected a window of 6 frames before each onset/offset and 6 frames after, so our segments are of 13 frames. Negative samples are randomly selected as segments of different length that do not contain positive labels. Fig. 2 (e) shows an example of onset detector scores (green dotted line) and offset detector scores (purple dotted line). As it can be seen in Fig. 2 (e) transition detectors are prone to noise and contain many false positives. However, a high response appears around the true onset, which allows CoT to refine the boundaries of detected segments with this partially correct information.

We linearly combine the transition and segment detection scores. Specifically, for any given segment $\mathbf{S}[s,e]$, we define the event score as $f_{\text{event}}(\mathbf{S}[s,e]) = \alpha f_{\text{seg}}(\mathbf{S}[s,e]) + \beta f_{\text{on}}(s) + (1-\alpha-\beta) f_{\text{off}}(e)$.

The combining parameters α and β indicate confidence on detectors and are learned by cross-validation. In practice, AUs with larger facial movements, *e.g.*, AU 12, tend to have larger values on the parameters for transition detectors.

Fig. 2 (d) shows the event score matrix of all possible segments in the input video. The largest score entry (△) provides a better estimate of the ground truth (△), that are superimposed, compared to the one obtained by the segment-level score matrix without transition scores (Fig. 2 (c)).

In Fig. 4, we show several qualitative results for various values of combining parameters, α and β. We use the same video sequence example as in Fig. 2, where onset and offset detection scores and other related information can be seen. In the figures, ground truth segment is marked △, while maximum score segments are marked with. Fig. 4 (a) shows f_{seg} alone. Fig. 4 (b) and c) show s effects of adding onset and offset detection s into the score matrix. Fig. 4 (d) shows a n example of result with appropriate combining parameters, while Fig. 4 (e) shows a n undesirable result when giving the noisy transition (onset) detection too many weights. Comparing among sub-figures in Fig. 4 reveals the importance of both the transition task and appropriate parameter choices.

To detect multiple AU events in a given video, we apply dynamic programming (DP) [56] to the event score matrix. Recall that the original DP solution [56] could return a long segment that merged multiple events as a long event. However, using the transition score provides more accurate information about where the true boundaries are, and CoT avoids this under-segmentation problem.

4. An efficient implementation of CoT

In this section, we describe an efficient implementation of CoT. As discussed in Section 3.3, DP can be used to segment AU events in image sequence by finding a global optimal segmentation. However, DP is computationally expensive to run it in large scale, especially when the sequences are long and relatively small portion of frames contain AUs. This section proposes an efficient temporal detection algorithm using the branch-and-bound (B&B) algorithm rather than the exhaustive DP. We follow a similar approach as Chu et al. [57].

The main intuition behind B&B is to avoid evaluating the segments where no AUs are possible. To search for all the segments with positive segment detection score in a image sequence, we parameterize the possible set of segments as $[s_{low}, s_{high}, e_{low}, e_{high}]$, where s_{low} and s_{high} are smallest and largest values for the indexes of the segment's start. Similarly, e_{low} and e_{high} are the smallest and largest values of for the indexes of the segment's end. Thus, for an image sequence from frame 1 to n, the set of all possible positions of a segment is represented as $[1, n, 1, n]$. With the parameterization, we explain below the two steps of branching and bounding in the B&B approach.

4.1. Branching

The branching step splits a large candidate set into disjoint subsets. We first compare the intervals between s_{low} to s_{high} and e_{low} to e_{high}. Then we select the larger interval and split it into halves. For example, if e_{low} to e_{high} is a larger interval, the candidate set $R = [s_{low}, s_{high}, e_{low}, e_{high}]$ is split into two subsets: $R_1 = [s_{low}, s_{high}, e_{low}, \lfloor \frac{e_{low}+e_{high}}{2} \rfloor]$ and $R_2 = [s_{low}, s_{high}, \lfloor \frac{e_{low}+e_{high}}{2} \rfloor + 1, e_{high}]$,

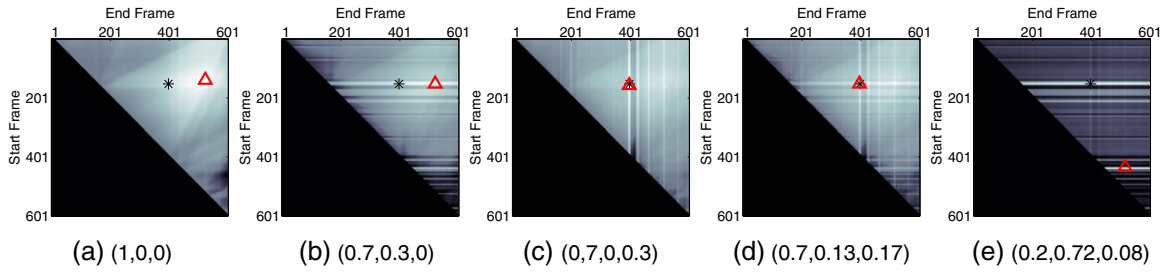

Fig. 4. Event scores for various values of combining parameters. The three numbers in parentheses stand for $(\alpha, \beta, 1-\alpha-\beta)$, which are weights for f_{seg}, f_{on} and f_{off} respectively. Transition detection scores and other related information can be seen in Fig. 2.

where $\lfloor \cdot \rfloor$ denotes to round towards the nearest smaller or equivalent integer.

4.2. Bounding

The bounding step calculates the upper bound of segment detection score for each segments set $R = [s_{low}, s_{high}, e_{low}, e_{high}]$. To improve the speed of the bounding process, we modify two aspects of the segment-level feature representation. First, we do not divide the segment into three sub-segments but compute the BoW using the whole segment. Second, we do not use the statistics on the detection score (*e.g.* minimum, maximum, median), but use the mean of the frame detection score.

Please note that these two modifications to segment-level feature representation are applied in both model training and segment searching, and we separately train a segment-level SVM model dedicated to CoT with B&B efficient search.

After these two modifications, it is straightforward to rewrite the segment detection score as a sum of positive and negative terms: $f_{\text{seg}}(\cdot) = f^{+}_{\text{seg}}(\cdot) + f^{-}_{\text{seg}}(\cdot)$, where $f^{+}_{\text{seg}}(\cdot)$ sums the prediction values from frames with positive contribution whereas $f^{-}_{\text{seg}}(\cdot)$ adds the negative ones. An upper bound of the segment set R can be written as: $\hat{f}_{\text{seg}}([s_{low}, s_{high}, e_{low}, e_{high}]) = f^{+}_{\text{seg}}(\mathbf{S}[s_{low}, e_{high}]) + f^{-}_{\text{seg}}(\mathbf{S}[s_{high}, e_{low}])$. This upper bound fulfills the two necessary conditions listed in Ref. [58]. First, it is larger or equal to the maximal segment score among all possible segments in the segment set. Second, it is exact when there is only one segment in the segment set.

With the branch and bound procedures explained above, we can perform efficient search for AU segments on a given image sequence. The procedure is shown in Algorithm 1. During the search, we maintain a priority queue P of candidate segment sets in given image sequence with n frames. The top state R_{top} is retrieved as the segment set that has the maximal upper bound $\hat{f}_{\text{seg}}$ in P. We repeatedly split (Branching) R_{top} into disjoint subsets and retrieve new top state R_{top}. This iteration ends when the top state R_{top} only contains one single segment $\mathbf{S}[s, e]$. If $f_{\text{seg}}(\mathbf{S}[s, e])$ is larger than 0, we first output $\mathbf{S}[s, e]$ as a detected AU segments, then re-initialize priority queue P and repeat the search. One way to re-initialize priority queue P is to remove all the detected AU segments from search range. Suppose that k AU segments are detected so far, denoted as $\{\mathbf{S}[s_i, e_i]\}_{i=1}^{k}$, $1 \le s_1 < e_1 < s_2 < e_2 \ldots < s_k < e_k \le n$. We can re-initialize P with states $[1, s_1 - 1, 1, s_1 - 1], [s_1 + 1, e_2 - 1, s_1 + 1, e_2 - 1], \ldots, [e_k + 1, n, e_k + 1, n]$. Any segment set that has $s_{low} > s_{high}$ or $e_{low} > e_{high}$ is removed from P. The search is repeated until the segment score of detected segment $f_{\text{seg}}(\mathbf{S}[s, e])$ is less than or equal to 0.

The branch-and-bound search detects AU segments $\{\mathbf{S}[s_i, e_i]\}_i$ with positive segment scores. To improve the boundary accuracy of detected segments, we perform a local search centered at each starting frame s_i and ending frame e_i, by using the transition detectors described in Section 3.3. Within a predefined radius (13 frames in our experiment) at each transition frame, the frame with maximal transition score is set as the final transition frame.

5. Experiments

We evaluated CoT on four datasets, the extended Cohn – Kanade (CK+) [33], GEMEP-FERA (FERA) [7], RU-FACS [11] and Sayette Group Formation Task (GFT) [59].

5.1. Experimental settings

This section describes the feature extraction methods, the training/test setup and the methods used for comparison.

5.1.1. Datasets

CK+ contains 593 facial expression sequences from 123 participants.

Most of them are posed facial expressions, while a small portion contains non-posed facial expressions.

Sequences vary in duration between 4 and 71 frames and the temporal structure of facial movements is predetermined. Each sequence begins with a neutral face and ends at peak intensity. Increases in AU intensity are monotonic.

Pose is frontal with relatively little head motion.

In FERA, we used the image sequences from the FERA training set of 87 portrayals from 7 trained actors.

Algorithm 1. Branch-and-bound search.

Data: Image sequence from frame 1 to n
Result: Detected AU segments $\{\mathbf{S}[s_i, e_i]\}_i$
Initialization:
Initialize priority queue P with single state $R = [s_{low}, s_{high}, e_{low}, e_{high}] = [1, n, 1, n]$;
Set the top state R_{top} as R;
repeat
  while R_{top} *contains more than one segment* **do**
    Split R_{top} into R_1 and R_2;
    Push R_1 and R_2 into P;
    Retrieve R_{top} from P, which is the state with maximal upper bound $\hat{f}_{\text{seg}}$;
  end
  Output R_{top} as a detected AU segment $\mathbf{S}[s, e]$, where $s = s_{low} = s_{high}, e = e_{low} = e_{high}$.
  if $f_{seg}(\mathbf{S}[s, e]) > 0$ **then**
    Push $\mathbf{S}[s, e]$ into $\{\mathbf{S}[s_i, e_i]\}_i$;
    Re-initialize priority queue P by removing all detected segments $\{\mathbf{S}[s_i, e_i]\}_i$;
    Retrieve R_{top} from P.
  end
until $f_{seg}(\mathbf{S}[s, e]) \le 0$;

Please note that although we only use image sequences in our experiment, FERA dataset also contains speech data.

Average duration is a little longer than 60 frames. AUs occur during emotional speech and have multiple apexes. Increases in AU intensity are not necessarily monotonic. Pose is primarily frontal with small to moderate change in head movement.

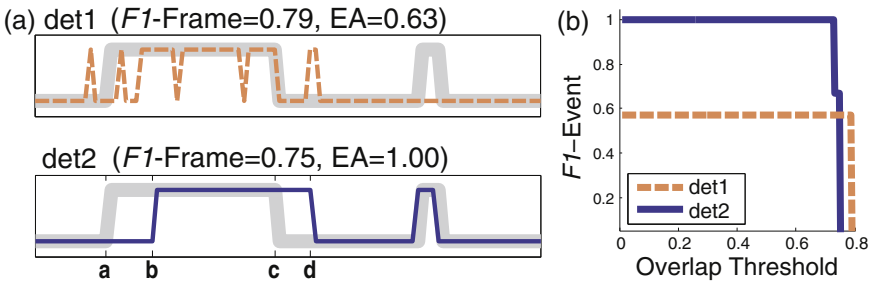

Fig. 5. Two synthetic detections for the metrics *F*1-Frame, Event Agreement (EA) [66] and *F*1-Event. (a) shows ground truth (gray thick line) and two detections (thin lines). In *F*1-Frame, `det1` scores higher although it has multiple false positives and misses a whole event. EA favors `det2` as it is more desirable in detecting AU events. In (b), *F*1-Event curve reflects boundary misalignment in `det2`, which is ignored in Event Agreement.

Table 1
*F*1-Frame on **CK+** dataset.

	Frame				Seg	CoT		
AU	CLM[a]	EF	MKL	LF	JSC	F	FS	FST
1	75	64.0	64.9	66.1	53.6	66.5	73.9	**76.2**
2	75	61.0	73.2	57.1	64.6	72.0	74.2	**76.3**
4	73	67.4	64.8	76.6	62.5	69.2	77.0	**78.5**
6	70	60.3	**74.7**	71.3	63.8	72.8	66.4	70.3
7	60	50.7	62.2	58.5	43.2	52.6	61.8	**63.4**
12	78	81.9	84.1	82.7	80.8	85.5	81.7	**86.8**
15	75	63.1	71.2	**79.8**	54.9	73.1	72.3	71.0
17	77	76.6	**86.1**	76.4	75.3	82.6	83.2	85.9
Average	72.9	65.6	72.6	71.0	62.3	71.8	73.8	**76.1**
Overall	–	66.3	72.7	71.2	62.1	71.7	74.6	**77.0**

Highest scores in each row are indicated with bold font.
[a] For each AU, Chew et al. [32] reported multiple results from different features. We selected the best ones and compute the average *F*1.

Table 2
*F*1-Frame on **FERA** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	57.6	61.1	54.9	50.9	55.9	62.5	**64.2**
2	49.4	54.4	52.6	49.0	49.8	56.0	**57.2**
4	43.6	45.4	**47.2**	44.3	36.8	46.7	46.6
6	62.3	67.0	72.8	70.1	66.0	72.1	**72.9**
7	61.3	65.1	67.0	66.0	61.5	65.5	**67.4**
12	71.5	75.4	77.9	76.8	70.8	77.0	**78.3**
15	38.9	44.3	37.5	33.3	38.0	44.6	**46.7**
17	30.1	36.7	34.9	30.7	33.4	**38.7**	38.6
Average	51.8	56.2	55.6	52.6	51.5	57.9	**59.0**
Overall	52.9	58.6	57.7	54.5	54.4	60.2	**61.4**

Highest scores in each row are indicated with bold font.

RU-FACS is more challenging than the other two datasets. It consists of facial behavior recorded during interviews of about 2 minutes duration. Participants show small to moderate pose variation and speech-related mouth movements. Compared with the above two datasets, RU-FACS is more natural in timing, much longer, and the AUs are at lower intensity. For technical reasons, we selected from 29 of 34 participants that were available to us with sequence length of about 7000 frames.

GFT dataset records real life social interactions among three-person groups in less constrained contexts. The videos were recorded by separate wall-mounted cameras faced each subject. We selected 50 image sequences of 50 subjects. Each image sequence has around 5400 frames. The videos include moderate-to-large head rotation and frequent occlusion, as results of the subjects frequently turned toward and away from each other and drank beverage. The facial movements are spontaneous and unscripted.

5.1.2. Face registration

For the CK+ and RU-FACS datasets person-specific Active Appearance Model [60] tracking of 66 facial landmarks was available. For FERA and GFT datasets, we used the recently proposed supervised descent method [54][1] to track 49 landmarks. All tracked facial features points were registered to a reference face by a similarity transformation.

5.1.3. Features

At frame level we extracted the geometric features [34] and appearance features (SIFT descriptor) [13]. See Section 3 for segment-level features.

5.1.4. Training/test spit

We used a leave-one-subject-out strategy in the CK+ and FERA dataset s . In RU-FACS, in order to compare with previously published results, we split the subject list into 19 subjects for training and 10 subjects for test. For more details on the training/testing spilt see Ref. [8]. In GFT, we randomly split subjects into 10 parts. We iteratively use each part of subjects as test set and the remaining as training set.

5.1.5. Frame-based methods to compare

We compared with three approaches that fuse shape [34] and appearance features [13]. For all methods we used the Radial Basis Function (RBF) kernel for shape features and concatenated features, and the Histogram Intersection Kernel (HIK) for SIFT features. The first method, Early Fusion (EF) [40], fuses features by concatenating feature vectors into a longer vector. Because different features have different range values, we normalized them to have zero mean and unitary variance. The second method, Late Fusion (LF) [9] combines outputs from classifiers trained on different features. Because the strength of features varies drastically across AUs, weighted averaging was used to obtain late fusion result, the weights were estimated by cross-validation. The third method for comparison was Multiple Kernel Learning (MKL) [61] that jointly estimates the SVM parameters and weights the contributions of different features.

5.1.6. Segment-based methods to compare

For segment-based methods, we implemented the Joint Segmentation and Classification (JSC) [56]. Note that JSC can be seen as segment detection in CoT without the input of the frame-level detector. Comparing JSC and FS reveals the contribution of the frame-level detector to the segment-level detector. Temporal words were constructed for the shape and appearance features separately, and then two kinds of segment-level feature vectors were concatenated. We used a linear SVM for the JSC.

5.1.7. Sequence learning methods to compare

We also implemented a hybrid SVM/hidden Markov model (HMM) approach as one sequence learning method. For each AU, we train a frame-based SVM model by using both geometric and appearance features, and a HMM with two states (*i.e.* activation and inactivation). This hybrid approach has been successfully applied in facial expression analysis [49] and speech recognition [62]. The idea is to combine SVM's discriminative power and HMM's ability to model time.

Specifically, the emission probabilities of HMM are computed based on SVM output with Platt scaling [63]. The state transition probabilities and *a priori* probabilities of HMM are estimated from training data. Because of this, datasets that depict real life social

Table 3
Event Agreement on **FERA** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	40.2	52.3	22.6	56.2	49.5	**65.5**	65.3
2	49.5	49.2	25.1	66.4	42.5	63.6	**71.4**
4	29.4	29.0	33.9	**53.6**	39.4	49.6	48.9
6	45.7	53.8	42.9	67.5	51.7	**67.7**	64.6
7	38.4	47.4	61.1	63.4	45.7	57.8	**63.6**
12	56.4	65.0	67.8	73.6	70.2	78.1	**79.9**
15	32.6	37.7	14.8	38.6	35.7	46.7	**48.6**
17	29.7	40.6	25.2	53.0	42.7	**59.3**	58.1
Average	40.2	46.9	36.7	59.0	47.2	61.0	**62.5**
Overall	39.2	46.3	32.5	58.8	47.2	61.4	**62.9**

Highest scores in each row are indicated with bold font.

[1] www.humansensing.cs.cmu.edu/intraface .

Table 4
*F*1-Frame on **RU-FACS** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	27.5	46.1	23.1	43.8	43.8	45.8	**49.7**
2	38.1	34.2	38.3	42.8	33.4	**47.5**	47.1
4	15.5	17.8	24.6	35.4	24.7	35.4	**36.5**
6	47.8	54.1	50.7	50.5	46.2	53.5	**56.2**
12	63.4	72.5	70.6	68.7	69.9	73.4	**77.5**
14	19.0	38.4	23.0	53.2	41.2	57.7	**59.2**
15	26.8	42.4	32.0	34.1	29.0	38.0	**43.0**
17	37.1	38.3	**42.9**	38.9	29.2	40.5	42.5
Average	34.4	43.0	38.1	45.9	39.7	49.0	**51.5**
Overall	37.4	49.1	40.7	52.3	43.9	56.0	**58.4**

Highest scores in each row are indicated with bold font.

interactions are preferred. During test, the most likely AU state path of each input video is determined by a standard Viterbi algorithm. We conducted this hybrid method experiment on GFT dataset, and denoted it as HMM in our experiment.

Please note that a two-state HMM model is certainly not the most representative model for this task. In future work we plan to include HCRF-kind of models, which allow active or inactive class to have multiple (latent) states.

5.1.8. SVM

For the linear and single kernel SVM we used the LIBSVM [64] and for MKL the SimpleMKL [65]. We did standard grid-search on the cross-validation parameters (including the C on the SVM).

5.2. Evaluation metrics

We first reported results using conventional metrics such as *F*1-Frame score. However, we argue that for many applications the *F*1-Frame score is less meaningful than an event-based metric. Therefore results are also reported in newly proposed event-based metrics.

5.2.1. F1-Frame

*F*1-Frame is widely used (*e.g.*, [7]) in AU detection literature. It is defined as $F1\text{-Frame} = \frac{2 * FR * FP}{FR+FP}$, where *FR* is the frame-level recall and *FP* is the frame-level precision. *F*1-Frame ignores temporal information and fails to reflect event-based performance. As an illustration, a synthetic detection example on 100 frames is shown in Fig. 5. Two detections (`det1` and `det2`) are shown along with ground truth. Note that `det1` misses one event and generates multiple false positives, while `det2` detects the correct number of events and roughly recovers their temporal locations. However, *F*1-Frame of `det1` is 0.79 (recall $= \frac{26}{37} \approx 0.70$, precision $= \frac{26}{29} \approx 0.90$), which is higher than 0.75 of `det2`. With this example we want to illustrate that the *F*1-Frame metric can miss important information, and our argument is that several evaluation metrics should be used.

Table 5
Event Agreement on **RU-FACS** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	21.7	**56.7**	31.3	36.6	25.1	38.7	47.5
2	23.6	37.1	35.7	45.7	23.3	**53.1**	52.7
4	6.6	15.0	13.3	**36.4**	8.0	28.5	33.7
6	19.0	41.7	53.7	68.8	27.5	71.1	**71.2**
12	49.5	65.8	71.4	71.4	48.7	**75.9**	70.5
14	15.2	20.3	14.3	62.6	35.3	65.6	**68.6**
15	12.6	28.4	24.1	39.1	22.6	53.3	**59.7**
17	20.3	26.7	33.2	35.5	17.5	40.4	**44.1**
Average	21.1	36.5	34.6	49.5	26.0	53.3	**56.0**
Overall	20.1	38.4	32.2	49.8	25.0	53.1	**56.7**

Highest scores in each row are indicated with bold font.

Table 6
*F*1-Frame on **GFT** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	26.5	23.3	28.0	23.8	23.1	28.6	**30.0**
2	24.2	25.3	27.3	25.5	25.0	**33.7**	33.6
6	68.6	71.1	69.5	71.9	70.4	**74.0**	73.7
7	68.6	69.3	69.4	69.7	68.4	**73.2**	73.1
10	69.8	72.6	71.9	70.4	68.7	**73.3**	73.3
11	41.4	39.8	41.2	40.2	39.7	41.4	**43.0**
12	69.6	71.6	72.1	68.3	67.0	73.5	**73.6**
14	69.6	**75.5**	62.5	65.6	64.6	66.5	66.9
15	25.4	30.2	29.0	30.5	29.8	**33.6**	33.3
17	39.2	46.4	45.3	36.1	36.8	42.1	**47.6**
23	31.4	30.6	**33.7**	30.3	30.2	29.4	29.6
24	29.8	28.8	32.5	28.2	27.0	32.4	**34.3**
1 + 2	21.2	21.6	23.9	22.7	21.5	29.1	**33.7**
6 + 7	67.2	66.8	64.0	66.9	65.6	**70.5**	70.3
12 + 14	64.8	68.5	67.4	67.1	65.6	72.5	**72.9**
Average	47.8	49.4	49.2	47.8	46.9	51.6	**52.6**
Overall	54.4	58.7	56.4	56.0	54.5	59.9	**60.2**

Highest scores in each row are indicated with bold font.

In AU detection, the numbers of positive samples and negative samples are often highly imbalanced. This is known as the skew problem. In GFT dataset, because the videos record facial expression from real-life conversations, the skew problem is particularly severe. To address this problem, besides the metrics described above, we also reported results using a newly proposed frame-level F1 score: skew-normalized F1 score [67], denoted as *F*1-Norm. *F*1-Norm is computed from a weighted confusion matrix that balancing the number of positive and negative frames. Reporting results using both standard and skew-normalized *F*1 minimized confounds from highly imbalanced dataset, as suggested in Ref. [67]. We will refer to the normalized *F*1 as *F*1-Norm.

5.2.2. Event Agreement

To model the event-based performance, a metric called Event Agreement (EA) was proposed in Ref. [66].

EA is defined as the percentage of agreed events between two annotations, which is: $\textit{Event Agreement} = \frac{\textit{Total Number of Agreed Events}}{\textit{Total Number of Identified Events}}$.

If an event in one annotation has overlap with other events from the other annotation, this event is called an agreed event.

In AU detection scenario, ground truth sequence and detection label sequence are two annotations, and EA measures the percentage

Table 7
Event Agreement on **GFT** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	19.4	19.1	19.3	40.1	18.8	36.4	**43.3**
2	20.6	22.1	21.7	35.6	24.1	35.8	**36.0**
6	53.3	48.2	47.6	68.2	48.2	68.6	**72.5**
7	47.0	51.2	45.3	65.6	49.5	69.1	**70.8**
10	57.6	53.8	61.2	74.5	63.0	72.2	**75.5**
11	32.7	34.1	37.3	40.6	33.8	39.2	**41.2**
12	52.5	54.7	53.9	71.7	60.2	73.7	**76.0**
14	69.6	65.5	68.9	**74.9**	74.4	73.6	73.8
15	24.3	27.3	27.5	39.6	26.5	**41.6**	41.0
17	49.4	**51.2**	45.4	42.0	47.5	42.3	45.4
23	35.2	33.6	37.2	**38.6**	33.1	36.8	37.5
24	29.3	29.0	30.3	42.0	29.4	45.2	**47.4**
1 + 2	15.4	22.7	21.9	34.9	22.8	36.0	**38.7**
6 + 7	52.4	48.8	65.1	66.2	50.5	69.8	**71.0**
12 + 14	56.3	53.2	49.3	69.1	57.1	72.0	**74.2**
Average	41.0	41.0	42.1	53.6	42.6	54.2	**56.3**
Overall	41.2	43.0	43.0	55.5	43.5	55.3	**57.1**

Highest scores in each row are indicated with bold font.

Table 8
*F*1-Norm on **GFT** dataset.

	Frame			Seg	CoT		
AU	EF	MKL	LF	JSC	F	FS	FST
1	59.8	53.8	**65.0**	55.9	56.0	60.7	62.0
2	**55.1**	50.5	48.7	51.7	51.7	52.4	52.4
6	74.3	78.7	76.6	79.3	78.0	**80.4**	79.4
7	73.2	73.7	74.3	74.0	72.8	**76.4**	76.1
10	73.3	**75.8**	75.1	72.9	71.2	75.4	75.0
11	59.9	54.8	51.5	55.0	54.7	58.6	**62.9**
12	77.1	77.6	**78.9**	73.3	72.0	77.4	77.3
14	64.5	**70.0**	59.1	62.4	61.3	63.5	64.1
15	**56.3**	52.0	50.1	49.2	52.3	48.7	48.7
17	47.6	56.5	55.1	42.2	43.2	48.5	**56.9**
23	54.9	43.2	**56.7**	42.0	42.9	38.9	39.1
24	52.4	48.9	**56.9**	45.8	45.8	51.5	55.4
1 + 2	**61.0**	50.4	53.3	50.1	50.5	53.4	60.7
6 + 7	74.7	75.5	67.6	75.1	73.9	**76.9**	76.1
12 + 14	72.1	77.4	78.0	74.8	73.4	**78.4**	78.4
Average	63.7	62.6	63.1	60.2	60.0	62.7	**64.3**
Overall	68.1	**71.0**	68.3	66.8	65.8	69.2	69.6

Highest scores in each row are indicated with bold font.

of events that are correctly detected (overlapped with ground truth events).

For example, in the `det2` (bottom figure of Fig. 5 (a)), there is an overlap between the ground truth event $[a, c]$ and the detected event $[b, d]$, therefore EA considers that the event is correctly detected (even if the overlap is minimal). In this case, EA for `det2` is $\frac{2+2}{2+2} = 1$. This is because, considering the thick line as ground truth two events are correctly detected (assuming a minimal overlap). Then, considering the thin line as ground truth two events are correctly detected. The EA is the ratio of events detected considering each of the signal as ground truth over the total number of events (in the two signals). For `det1` (top figure in Fig. 5 (a)), the EA is $\frac{1+4}{2+6} \approx 0.63$.

5.2.3. *F*1-*Event curve*

A major problem for EA to be used as a measure for AU detection, is that a single frame of overlap between the detected AU event and ground truth is considered as an event agreement. For example, in Fig. 5, although `det2` gets full score in EA, it is not a perfect detection, especially in transition regions. To address this issue, we propose a novel event-based metric: *F*1-Event $= \frac{2 \bullet ER \bullet EP}{ER+EP}$, where Event-based Recall (*ER*) is the ratio of correctly detected events over the true events, while the Event-based Precision (*EP*) is the ratio of correctly detected events over the the detected events. Unlike EA, *F*1-Event considers that there is an event agreement if the overlap is above a certain threshold, which can be set depending on specific applications. For the purpose of comparison the *F*1-Event curve is generated by varying the overlap threshold. For example, in Fig. 5 (b), *F*1-Event curves for `det1` and `det2` are shown. `det2` for most thresholds has higher *F*1-Event score, except in the regions with extremely high threshold. This is because detected events of `det1` are shorter and once they are agreed they tend to get a high overlap ratio. It is interesting to note that when the overlap threshold is zero, *F*1-Event is very close to EA, as they are both "averaging" *ER* and *EP*.

5.3. *Results*

We reported results across all evaluation metrics (*F*1-Frame, EA, *F*1-Event). We also reported intermediate results, F (frame detection result) and FS (frame and segment detection s without transition), in order to analyze the contribution of each task. To show the detection performance for all AUs, we reported the *Average* and *Overall F*1 scores. The *Average F*1 corresponds to the mean value of F1 scores for all AUs. The *Overall F*1 was calculated from an overall confusion matrix. The overall confusion matrix was computed by summing confusion matrices of all AUs. By doing so, we implicitly assigned larger weights to the AUs that appear more frequently. Because CK+ does not contain complete AU events, event-based metrics (*i.e.*, EA and *F*1-Event) were not used in CK+.

In addition, we tested efficient implementation of CoT on GFT dataset, which has relatively long image sequence durations. Thus we reported CoT results using both DP and B&B segmentation techniques, denoted as CoT_BB and CoT_DP respectively.

5.3.1. *F*1-*Frame*

Results are shown in Table 1 (CK+), Table 2 (FERA), Table 4 (RU-FACS) and Table 6 (GFT). We also included the detection results on CK+ reported by Chew et al. [32] using Constrained Local Models (CLM). First, the final result of CoT (FST) outperforms all the other methods. In terms of *overall F*1-Frame, on CK+, the difference between FST and the second best method (MKL) is 4.3; on FERA, the difference between FST and the second best method (MKL) is 2.8; on CK+, the difference between FST and the second best method (JSC) is 6.1. Second, in our experiments the methods using multiple features did not necessarily perform better than the methods using single feature. This might be due to the redundancy of the features and possible normalization artifacts. For frame-based methods, MKL

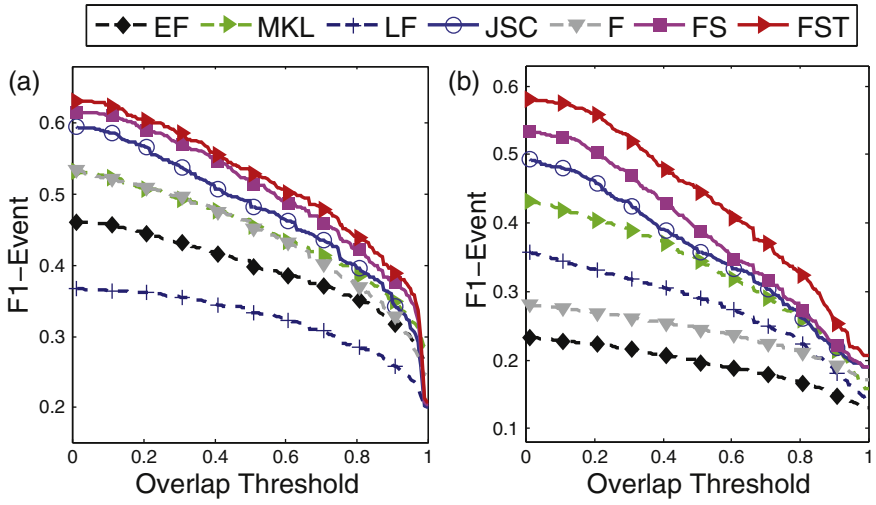

Fig. 6. Overall *F*1-Event on (a) FERA and (b) RU-FACS dataset s . Overlap threshold varies from 0.01 to 1. Solid and dotted lines denote segment- and frame-based methods, respectively.

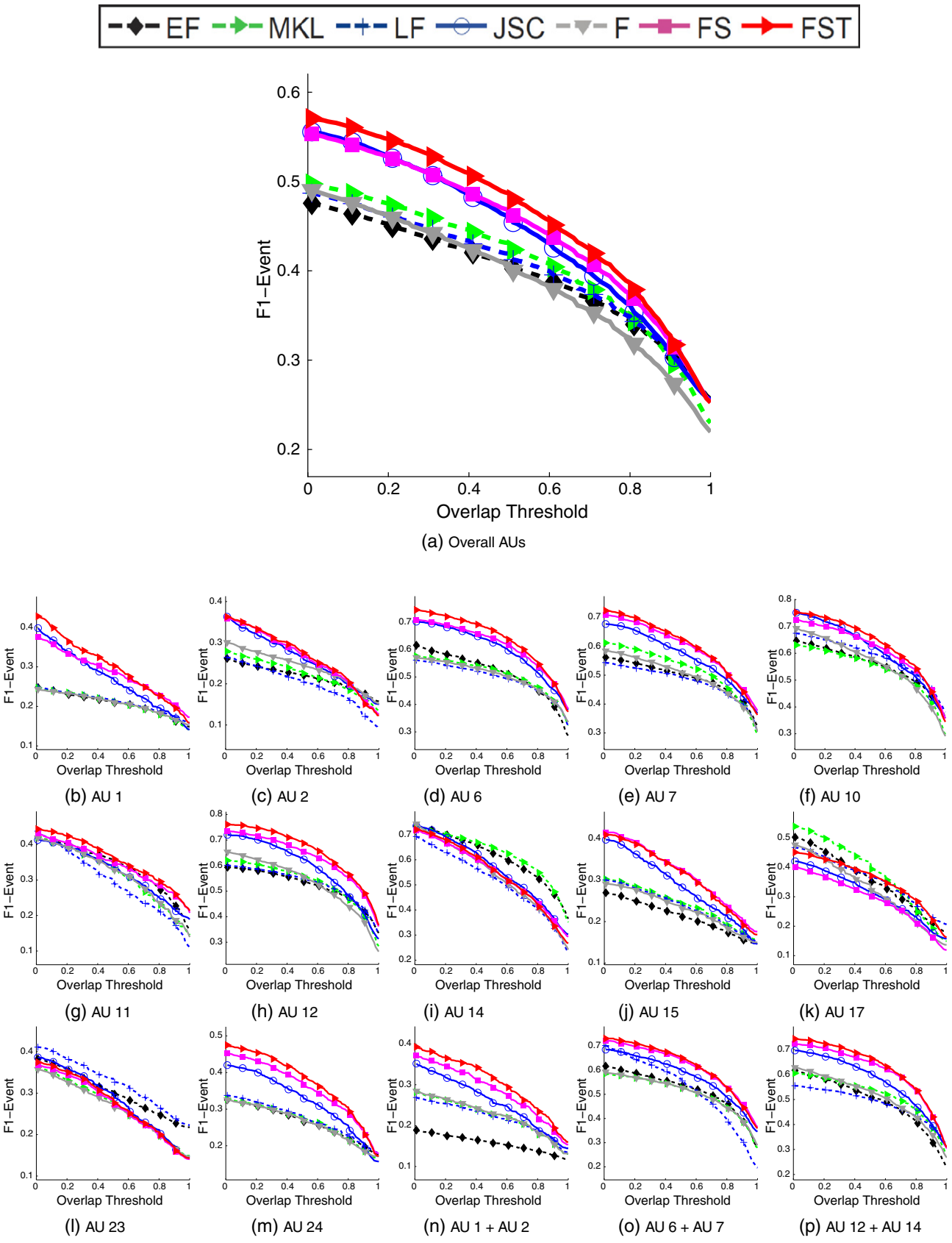

Fig. 7. *F*1-Event curves on GFT dataset. Overall *F*1-Event curves are shown in (a) and curves for each AU are shown in (c – p). (b) shows performance of CoT with efficient implementation (CoT_BB). Overlap threshold varies from 0.01 to 1. Solid and dotted lines denote segment- and frame-based methods, respectively.

Table 9
Comparison between CoT_DP, CoT_BB and HMM on **GFT** dataset.

	*F*1-Frame			EA			*F*1-Norm		
AU	DP	BB	HMM	DP	BB	HMM	DP	BB	HMM
1	**30.0**	28.0	27.5	43.3	**47.5**	42.4	**62.0**	61.3	60.7
2	**33.6**	31.8	25.3	36.0	36.7	**42.2**	52.4	51.2	**55.4**
6	73.7	**74.1**	67.9	72.5	**78.5**	71.0	79.4	**81.4**	73.2
7	73.1	**74.6**	68.0	70.8	**73.7**	68.4	76.1	**78.2**	72.2
10	73.3	**74.7**	69.4	**75.5**	73.7	72.0	75.0	**77.1**	72.5
11	**43.0**	42.3	40.5	41.2	44.6	**44.6**	62.9	**63.6**	58.1
12	**73.6**	73.1	70.6	**76.0**	75.6	72.8	77.3	**78.6**	77.6
14	66.9	**69.0**	67.3	**73.8**	65.3	73.7	64.1	**65.7**	62.8
15	**33.3**	30.7	25.3	41.0	33.7	**41.2**	48.7	43.7	**55.3**
17	**47.6**	44.6	38.7	**45.4**	32.1	42.4	**56.9**	54.0	47.1
23	29.6	25.5	**30.0**	37.5	24.8	**50.5**	39.1	32.7	**52.0**
24	**34.3**	27.8	30.9	**47.4**	35.4	45.2	55.4	46.0	**55.6**
1 + 2	**33.7**	32.4	22.4	**38.7**	37.3	37.0	60.7	57.9	**62.5**
6 + 7	70.3	**71.1**	67.7	71.0	71.3	**71.6**	76.1	**79.1**	74.7
12 + 14	**72.9**	72.1	65.5	**74.2**	72.8	66.9	78.4	**79.8**	72.7
Average	**52.6**	51.5	47.8	**56.3**	53.5	56.1	**64.3**	63.3	63.5
Overall	60.2	**60.5**	54.1	**57.1**	53.4	56.1	69.6	**71.1**	67.2

Highest scores in each row are indicated with bold font.

is the most stable and EF typically gets the lowest scores (even lower than F that only uses SIFT features). The results using *F*1-Norm are shown in Table 8 for GFT.

5.3.2. Event Agreement

Results are shown in Table 3 (FERA), Table 5 (RU-FACS) and Table 7 (GFT). First, the advantage of segment-based methods (JSC, FS, FST) over frame-based methods (EF, MKL, LF, F) is clear. On FERA and RU-FACS, mean *overall* EA differences between segment-based and frame-based methods are 19.8 and 24.3, respectively. Second, FS consistently outperforms JSC. This shows how frame detection helps in segment detection stage. Third, because EA does not consider the overlap ratio, the performance improvement done by using the transition task is not well reflected with the metric. This explains why under EA the advantage of FST over FS is insignificant, and in some cases when transition detection is highly noisy, FS is even better.

5.3.3. F1-Event curve

For FERA and RU-FACS, results are shown in Fig. 6 (a) and (b). First, the top three lines on both datasets are segment-based methods (solid lines), which best shows segment-based method's advantage in detecting AU events. Second, because most AU events in RU-FACS are complete, opposed to lots of incomplete events in FERA, RU-FACS contains more AU transitions. Hence transition detection (only in FST) plays a more important role, which is revealed by the gap between the top two curves. In some cases in FERA, false transition detection even results in worse FST results than FS.

For GFT dataset, we shown *F*1-Event curves of overall and each AUs in Fig. 7. Fig. 7 (c – p) show curves for each AUs. Most AUs show similar pattern with overall figure Fig. 7 (a). However, two notable exceptions are AU 17 and AU 23. One reason behind this is that GFT dataset contains lots of short events of AU 17 and AU 23. In fact, the median event length for AU 17 and AU 23 is 9 and 10, which are respectively the shortest and second shortest among all AUs. Meanwhile, the median event length for all AUs is 22. This shows that when the AU event length is short, CoT's performance becomes close to or even worse than frame-based methods.

Across the above three metrics, CoT (FST) consistently performed the best among all AU detection methods for comparison. An increasingly performance improvement within CoT was observed while

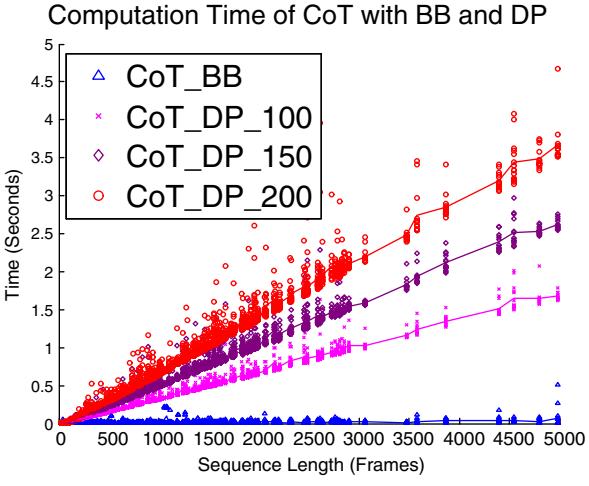

Fig. 8. Time for CoT_BB and CoT_DP_100, CoT_DP_150, CoT_DP_200 with maximum length of 100, 150 and 200 respectively. Dotted lines are added to show how computation time increases with the length of the sequence. DP scales linearly, while B&B is mostly constant.

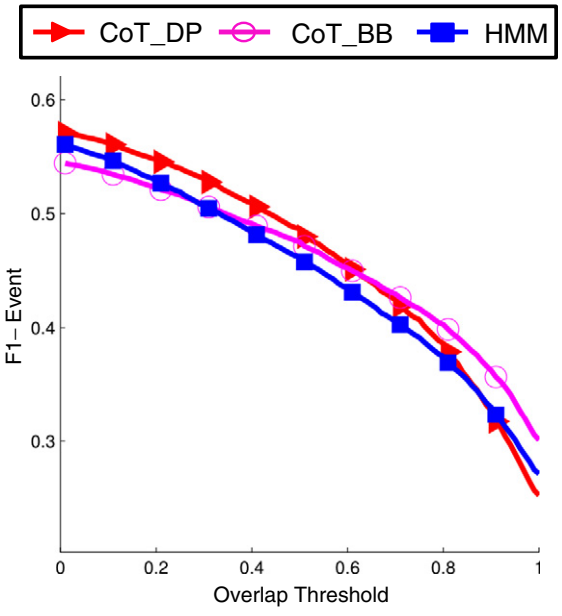

Fig. 9. Overall *F*1-Event of CoT_DP, CoT_BB and HMM on GFT dataset. Overlap threshold varies from 0.01 to 1.

new task(s) being integrated. This improvement is more obvious on RU-FACS and GFT where more complete AU events were present.

5.3.4. Three approaches in segmenting sequence

In our experiment, we included three dynamic approaches for modeling time of AU events and segmenting test sequences, which are DP, B&B and HMM. In Table 9 and Fig. 9, we show performance of the three approaches. Overall, CoT_DP and CoT_BB perform better, while these three methods generate comparable results.

It is interesting to note that in terms of *F*1-Event curve (Fig. 9), CoT_DP performed better when the overlap threshold is small (between 0 and 0.6), while CoT_BB performed better when the overlap threshold is higher (0.6 to 1). One reason of CoT_DP's performance drops quicker in high overlap area is the limitation on the length of the search segments. Because CoT_DP uses exhaustive search, in our implementation of CoT_DP, a maximum search length has to be set for computational feasibility. The maximum search length was set to be 200 frames through our experiment. Meanwhile, CoT_BB does not have this limitation.

In addition, to demonstrate the efficiency, we reported the computation time of CoT_BB (the efficient implementation) and CoT_DP in Fig. 8. The results were computed using a standard laptop with 2.8 GHz dual core CPU and 4 G B RAM. The AU event searches are performed on sequences with different lengths in GFT dataset. As can be seen, the computation time for CoT_DP increases linearly with sequence length, while computation time of CoT_BB is invariant of sequence length. To make this situation more clear, we show CoT_DP with three different maximum search segment lengths.

6. Conclusion

Most previous approaches to AU detection are concerned with detecting occurrence in single frames and thus ignore the coherence of AU events that can span multiple frames. We proposed a method to detect facial AU events from image sequences. In our method, three complementary detection tasks are sequentially combined. Experiments on four datasets that differ in complexity show that our method outperformed state-of-the-art alternatives in each case. The improved AU detection relative to state-of-the-art was achieved by combining tasks rather than by increasing computational complexity. The idea of using a cascade of detection tasks of varying granularity is not limited to facial AU detection. Future work could apply this approach to other detection applications with temporally continuous data, such as human gesture detection.

Acknowledgment

Research reported in this publication was supported in part by the National Institute of Mental Health of the National Institutes of Health under award number MH096951 and the National Science Foundation under the grants RI-1116583, IIS 1418026 and 1418520-L. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or the National Science Foundation. X. Ding and Q. Wang are supported by Specific Medical Science Fund of Technological Innovation and Achievements Transformation of Jiangsu Province under Grant No. BL2012025, and NSF of China under Grant No. 91330109.

References

[1] Z. Ambadar, J.F. Cohn, L.I. Reed, All smiles are not created equal: morphology and timing of smiles perceived as amused, polite, and embarrassed/nervous, J. Nonverbal Behav. 33 (1) (2009) 17–34.
[2] C.E. Fairbairn, M.A. Sayette, J.M. Levine, J.F. Cohn, K.G. Creswell, The effects of alcohol on the emotional displays of Whites in interracial groups, Emotion 13 (3) (2013) 468–477.
[3] P. Ekman, W.V. Friesen, J.C. Hager, Facial Action Coding System: Research Nexus, Network Research Information, Salt Lake City UT, 2002.
[4] J.F. Cohn, Z. Ambadar, P. Ekman, Observer-Based Measurement of Facial Expression With the Facial Action Coding System, Oxford University Press Series in Affective Science, New York NY: Oxford University, 2007.
[5] P. Ekman, E. Rosenberg, What the Face Reveals, second ed., Oxford, New York, NY, 2005.
[6] A. Martinez, S. Du, A model of the perception of facial expressions of emotion by humans: research overview and perspectives, J. Mach. Learn. Res. 13 (2012) 1589–1608.
[7] M.F. Valstar, M. Mehu, B. Jiang, M. Pantic, K. Scherer, Meta-analysis of the first facial expression recognition challenge, IEEE Transactions IEEE Trans. Syst. Man Cybern. B 42 (4) (2012) 966–979.
[8] W.S. Chu, F. De la Torre, J.F. Cohn, Selective transfer machine for personalized facial action unit detection, CVPR, 2013.
[9] U. Tariq, T. Huang, Features and fusion for expression recognition — a comparative analysis, CVPR, 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6239229.
[10] M. Pantic, Expert system for automatic analysis of facial expressions, Image Vis. Comput. 18 (11) (2000) 881–905. http://linkinghub.elsevier.com/retrieve/pii/S0262885600000342.
[11] M.S. Bartlett, G.C. Littlewort, M.G. Frank, C. Lainscsek, I.R. Fasel, J.R. Movellan, Automatic recognition of facial actions in spontaneous expressions, J. Multimedia 1 (6) (2006) 22–35.
[12] T. Simon, M.H. Nguyen, F. De la Torre, J.F. Cohn, Action unit detection with segment-based SVMs, CVPR, 2010.
[13] Y. Zhu, F.D.e.l.a. Torre, J.F. Cohn, Y.-J. Zhan, Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior, IEEE Trans. Affect. Comput. (2011) 1–14. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5740840.
[14] G. Stratou, A. Ghosh, P. Debevec, L.-P. Morency, Exploring the effect of illumination on automatic expression recognition using the ICT-3DRFE database, Image Vis. Comput. 30 (10) (2012) 728–737. http://linkinghub.elsevier.com/retrieve/pii/S0262885612000169.
[15] G. Zhao, X. Huang, M. Taini, S.Z. Li, M. Pietikäinen, Facial expression recognition from near-infrared videos, Image Vis. Comput. 29 (9) (2011) 607–619. http://linkinghub.elsevier.com/retrieve/pii/S0262885611000515.
[16] J.F. Cohn, F. De. la Torre, Automated face analysis for affective computing, in: Handbook of Affective Computing, Oxford, New York, NY.
[17] D. McDuff, R.E.l. Kaliouby, T. Senechal, D. Demirdjian, R. Picard, Automatic measurement of ad preferences from facial responses gathered over the internet, Image Vis. Comput. (2014) 1–11. http://linkinghub.elsevier.com/retrieve/pii/S026288561400016X.
[18] G.C. Littlewort, M.S. Bartlett, K. Lee, Automatic coding of facial expressions displayed during posed and genuine pain., Image Vis. Comput. 12 (27) (2009) 1797–1803.
[19] Image. EEG for implicit affective tagging, S. Koelstra, I. Patras, Vision. Computing, 31 (2) (2013) 164–174. http://linkinghub.elsevier.com/retrieve/pii/S0262885612001825.
[20] M. Reale, P. Liu, L. Yin, Using eye gaze, head pose, and facial expression for personalized non-player character interaction, CVPR Workshops, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5981691.
[21] T. Wu, N.J. Butko, P. Ruvolo, J. Whitehill, M.S. Bartlett, J.R. Movellan, Multilayer architectures for facial action unit recognition, IEEE Trans. Syst. Man Cybern. B 42 (4) (2012) 1027–1038.
[22] S. Lucey, A.B. Ashraf, J.F. Cohn, Investigating spontaneous facial action recognition through AAM representations of the face, Face Recognition (2007) 275–286.
[23] G. Zhao, M. Pietikäinen, Boosted multi-resolution spatiotemporal descriptors for facial expression recognition, Pattern Recogn. Lett. 30 (12) (2009) 1117–1127. http://linkinghub.elsevier.com/retrieve/pii/S0167865509000695.
[24] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, D.N. Metaxas, Learning active facial patches for expression analysis, CVPR, 2012.
[25] K.Y. Chang, T.L. Liu, S.H. Lai, Learning partially-observed hidden conditional random fields for facial expression recognition, CVPR, 2009.
[26] L. Shang, Nonparametric discriminant HMM and application to facial expression recognition, CVPR, 2009. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5206509.
[27] Y. Tong, J. Chen, Q. Ji, A unified probabilistic framework for spontaneous facial action modeling and understanding, PAMI 32 (2) (2010) 258–273.
[28] F. De la Torre, T. Simon, Z. Ambadar, J.F. Cohn, FAST-FACS: a Computer-Assisted System to Increase Speed and Reliability of Manual FACS Coding, 2011.
[29] F. De la Torre, J.F. Cohn, Facial expression analysis, Visual Analysis of Humans: Looking at People, 2011, 377.
[30] M. Pantic, M.S. Bartlett, Machine analysis of facial expressions, Face Recognition 2 (8) (2007) 377–416.
[31] G. Sandbach, S. Zafeiriou, M. Pantic, L. Yin, Static and dynamic 3d facial expression recognition: a comprehensive survey, Image Vis. Comput. http://dx.doi.org/10.1016/j.imavis.2012.06.005.
[32] Others, Person-Independent Facial Expression Detection Using Constrained Local Models, 2011.
[33] The Extended Cohn – Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression, 2010.

[34] F. Zhou, F.D.e.l.a. Torre, J.F. Cohn, Unsupervised discovery of facial events, CVPR, 2010. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5539966.
[35] G.C. Littlewort, M.S. Bartlett, I. Fasel, J. Susskind, J. Movellan, Dynamics of facial expression extracted automatically from video, Image Vis. Comput. 24 (6) (2006) 615–625.
[36] C. Shan, S. Gong, P.W. Mcowan, Facial expression recognition based on local binary patterns : a comprehensive study, Image Vis. Comput. 27 (6) (2009) 803–816. http://dx.doi.org/10.1016/j.imavis.2008.08.005.
[37] B. Jiang, M.F. Valstar, M. Pantic, Action unit detection using sparse appearance descriptors in space-time video volumes, AFGR, 2011.
[38] 231-245, X. Zhao, E. Dellandréa, J. Zou, L. Chen, A unified probabilistic framework for automatic 3d facial expression analysis based on a Bayesian belief inference and statistical feature models, 2013, http://linkinghub.elsevier.com/retrieve/pii/S0262885612001813.
[39] A. Savran, B. Sankur, M.T.a.h.a. Bilge, Regression-based intensity estimation of facial action units, Image Vis. Comput. 30 (10) (2012) 774–784. http://linkinghub.elsevier.com/retrieve/pii/S0262885611001326.
[40] U. Tariq, K.-H. Lin, Z. Li, X. Zhou, Z. Wang, V. Le, T.S. Huang, X. Lv, T.X. Han, Emotion recognition from an ensemble of features, AFGR, 2011. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5771365.
[41] T.R. Almaev, M.F. Valstar, Local Gabor binary patterns from three orthogonal planes for automatic facial expression recognition, 2013, 356–361. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6681456.
[42] Decision level fusion of domain specific regions for facial action recognition, B. Jiang, B. Martinez, M. Valstar, M. Pantic, ICPR, 2014. http://www.braismartinez.com/media/documents/2014icprjiang.pdf.
[43] S. Chen, Y. Tian, Q. Liu, D.N. Metaxas, Recognizing expressions from face and body gesture by temporal normalized motion and appearance features, Image Vis. Comput. 31 (2) (2013) 175–185. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5981880.
[44] G. Sandbach, S. Zafeiriou, M. Pantic, D. Rueckert, Recognition of 3d facial expression dynamics, Image Vis. Comput. 30 (10) (2012) 762–773. http://linkinghub.elsevier.com/retrieve/pii/S0262885612000157.
[45] B. Jiang, M.F. Valstar, B. Martinez, M. Pantic, A dynamic appearance descriptor approach to facial actions temporal modeling., IEEE Trans. Cybern. 44 (2) (2014) 161–174.
[46] R. Walecki, O. Rudovic, V. Pavlovic, M. Pantic, Variable-state latent conditional random fields for facial expression recognition and action unit detection, AFGR, 2015.
[47] M. Pantic, I. Patras, Detecting facial actions and their temporal segments in nearly frontal-view face image sequences, IEEE Int'l Conf. on Systems, Man and Cybernetics, 4, 2006. pp. 3358–3363.
[48] M. Pantic, I. Patras, Dynamics of facial expression : recognition of facial actions and their temporal segments, IEEE Trans. Syst. Man Cybern. B 36 (2) (2006) 433–449.
[49] M.F. Valstar, M. Pantic, Fully automatic recognition of the temporal phases of facial actions, IEEE Trans. Syst. Man Cybern. B 42 (1) (2012) 28–43.
[50] A dynamic texture-based approach to recognition of facial actions, S. Koelstra, M. Pantic, I. Patras, their temporal models, PAMI 32 (11) (2010) 1940–1954. http://www.ncbi.nlm.nih.gov/pubmed/20847386.
[51] Combined Support Vector Machines and Hidden Markov Models for Modeling Facial Action Temporal Dynamics, 2007.
[52] Kernel Conditional Ordinal Random Fields for Temporal Segmentation of Facial Action Units, 2012.
[53] R. Caruana, Multitask learning, Mach. Learn. 75 (1997) 41–75.
[54] X. Xiong, F.D.e.l.a. Torre, Supervised descent method and its applications to face alignment, CVPR, 2013. http://www.ri.cmu.edu/pub_files/2013/5/main.pdf.
[55] X. Wu, R. Srihari, Incorporating prior knowledge with weighted margin support vector machines, SIGKDD, ACM Press 2004.
[56] M. Hoai, Z.-Z. Lan, F.D.e.l.a. Torre, Joint segmentation and classification of human actions in video, CVPR, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5995470.
[57] http://link.springer.com/chapter/10.1007/978 3-642-33765-9_27, W.-S. Chu, F. Zhou, F.D.e.l.a. Torre, Unsupervised temporal commonality discovery, ECCV, 2012.
[58] C.H. Lampert, M. Blaschko, T. Hofmann, Efficient subwindow search: a branch and bound framework for object localization, PAMI 31 (12) (2009) 2129–2142. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5166448.
[59] M.A. Sayette, K.G. Creswell, J.D. Dimoff, C.E. Fairbairn, J.F. Cohn, B.W. Heckman, T.R. Kirchner, J.M. Levine, R.L. Moreland, Alcohol and group formation: a multimodal investigation of the effects of alcohol on emotion and social bonding, 23 (8) (2012) 869–878. http://www.ncbi.nlm.nih.gov/pubmed/22760882.
[60] I. Matthews, S. Baker, Active appearance models revisited, IJCV 60 (2) (2004) 135–164.
[61] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, L. Prevost, 2012, Facial action recognition combining heterogeneous features via multikernel learning, IEEE Trans. Syst. Man Cybern. B 42 (4) 993–1005, http://www.ncbi.nlm.nih.gov/pubmed/22623430.
[62] Speech recognition with support vector machines in a hybrid system, S. Krüger, M. Schafföner, M. Katz, INTERSPEECH 1 (2005) 1–4. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Speech+Recognition+with+Support+Vector+Machines+in+a+Hybrid+System#0.
[63] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, http://www.tu-harburg.de/ti6/lehre/seminarCI/slides/ws0506/SVMprob.pdf.
[64] 27:1-27:27, C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011).
[65] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, Others, simpleMKL, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[66] Crowdsourcing Micro-Level Multimedia Nnnotations : The Challenges of Evaluation and Interface, 2012.
[67] L. Jeni, J.F. Cohn, F.D.e.l.a. Torre, Facing imbalanced data-recommendations for the use of performance metrics, 2013. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6681438.