A Hybrid System for On-line Blink Detection

Yijia Sun
Department of Computing
Imperial College London
Email: yijia.sun10@imperial.ac.uk

Stefanos Zafeiriou
Department of Computing
Imperial College London
Email: s.zafeiriou@imperial.ac.uk

Maja Pantic
Department of Computing
Imperial College London
Faculty of EEMCS
University of Twente
Email: m.pantic@imperial.ac.uk

***Abstract*—Automatic non-obtrusive deception detection is highly desirable because of its objectivity, accuracy and reliability. Eye blinking has been shown to be one of the informative non-verbal behavioural cues for solving this problem. Traditional blink recognition methods tend to use a tracker to extract static eye region images and classify those images as open and closed eyes to detect blinks. However, those recognition systems are frame based and do not incorporate temporal information. For this reason, they perform poorly as the tracker fails to detect eyes due to rapid head movement or occlusion. In this paper, we present an approach which combines Hidden Markov Models and Support Vector Machines to model the temporal dynamics of eye blinks and improve the blink detection accuracy.**

I. Introduction

Deception detection and determination of concealment-of-intent are very important areas of research as they have a wide range of applications. These include airport security check-points, border crossing stations, and other security screening points. Also thousands of people are treated on a daily basis for suicidal depression, schizophrenia, and eating disorders. The ability to deceive is considered to have been part of the evolution process, as being able to conceal your intent improved an individuals survival chances. For this reason, deception is an inevitable aspect of human interaction [1]. Given its pervasiveness, one might expect that humans would be adept at spotting deceit. However, several recent meta-analytic studies [2] have shown that professionals and lay people alike perform poorly at detecting deceit or concealed intent to do others harm, achieving detection accuracy averages that hover slightly above chance (54%). In addition, several studies in experimental psychology suggest that some of the visual behavioural signals can not be identified or tracked by the human eye because they are too subtle and fleeting to measure [3]. Hence, automated, unobtrusive monitoring and assessment of concealment-of-intent behaviours will form a valuable tool for all involved professionals. In addition to several popular non-verbal behavioural cues which have been adopted for deception detection (such as facial expression, body gestures, voice and verbal style), there is conclusive evidences showing that eye blinking behaviour is also related [4]. Thus, we employed eye blinking as cues in our research.

In recent decades, research regarding eye blink detection has been conducted. In [5], a method based on dual state tracking was presented which used intensity and edge information to distinguish closed and open eyes. An improved version of their techniques was later proposed in [6] in order to detect more states including open, narrow and closed. In particular, the system fed Gabor coefficients into a neural network and analysed eye-relevant action units (AU41, AU42 and AU43) to recognize eye state. In [7][8], the method localized eyes by applying motion analysis and it calculated the normalized cross correlation with pre-trained open-eye and closed-eye templates for each frame. It detected blinks by observing the waveform of the correlation scores and helped to classify voluntary and spontaneous blinks based on the blink duration. This method was applied in a real-time vision-based HCI system which was designed to help disabled people to interact with computer using voluntary blinks. Another technique, introduced in [9], clustered upper and lower eyelids after processing point-based motion pictures. The blink waveform was plotted through calculating the space between the upper and the lower eyelid. And a driver drowsiness detection system was developed based on this technique for security concerns. In [10][11], an approach using frame differencing and optical flow was introduced. It was shown that image flow analysis which contains both the magnitude and the direction of eyelid movement was more reliable than static appearance. In [12], an eye-blinking detection system was designed based on the analysis of the deformation of active contours that captured the eye. In [13], features produced by the application of a bank of Gabor filters were used for eye-blink detection. In [14], a detailed eye region model was used for blinking detection. Finally, a comparison between different features for open and closed eye detection was performed in [15]. Noticeably, these systems all require robust algorithms for eye tracking and may not perform well when the tracker can not detect the eye region correctly. A robust eye blink recognition system in unconstrained environment remains a challenging problem. Particularly, the aforementioned methods do not take temporal information into consideration, so they may produce erroneous results in frames in which the eye regions cannot be accurately identified.

To solve this problem, we propose a hybrid blink detection method which combines Hidden Markov Models and Support Vector Machines. The introduction of the temporal dynamics also enables our method to distinguish between the states of closed and half-open eyes. Therefore, in the line of previous work on temporal segmentation of AUs [16], [17], [18], [19], our method can segment blinks from continuous videos and detect four states according to four temporal segments of blinking: neutral (open-eye), onset (closing eye), apex (closed-eye) and offset (half-open eye). Several features were extracted from each frame and their performances were compared in testing process. Two hybrid models, the blink model and the non-blink model, were trained on image sequences with and without blinks respectively. Blinks were detected by comparing the two models' likelihoods, and durations were obtained through calculating the number of the apex states after decoding. A sliding window was further applied to enable on-line blink detection in real-time.

II. Hybrid System

This paper introduces an approach which models full temporal dynamics of eye blinking. This method comprises four main steps: extracting features for each frame from eye image sequences, classifying those features for each frame, training hybrid temporal models and applying a sliding window on the testing segment to enable the detection system to work in real-time.

HMMs can represent the temporal dynamics of eye blinking efficiently. The emission probabilities are usually estimated using mixtures of Gaussian probability density functions. These Gaussian mixtures suffer from poor discrimination because they are trained by likelihood maximization which assumes the model is correct. In contrast, SVM discriminates one class from the other one extremely well. Thus, a hybrid SVM-HMM model was exploited by our blink detection system.

Previous works were revolving around capturing transitions between open eyes and closed eyes. However, this is not able to fully describe eye blinking since blinks are usually more subtle and complex. Therefore, we employed four temporal states of blinking in our system: neutral (open-eye), onset (closing eye), apex (closed-eye) and offset (half-open eye). All the frames were labelled as 1,2,3 and 4 according to which state was appropriate.

A. Feature Extraction

We have exploited several popular features independently and compared their performance in our blink detection system: HOG (Histogram of oriented gradients), Gabor filter, LBP (Local Binary Pattern), optical flow and pixel intensity: HOG counts the distributions of gradient direction and edge orientation in each localized cell of a static image. It can be used as a spatial descriptor which can recognize eyelid position in order to detect blink. Another method which can also be used to monitor the position of eyelids is the Gabor Filter. It is a complex exponential modulated by a Gaussian function in the spatial domain. Generally, a Gabor filter bank is created by filters of five scales and four orientations. Another feature used is LBP, which compares each pixel with its neighbours in the cell of an image and gets an eight-digit output for each cell. The distribution of those outputs was calculated and applied for different eye state classification. Additionally, optical flow is employed, which captures the relative motion between consecutive frames which can indicate both magnitude and direction of the eyelid movement.

B. Pairwise SVM Classification

Once the features for all sets of image sequences have been extracted, a group of pairwise SVM classifiers were trained for the multi-class classification. For each pair of temporal segments, we used one pairwise classifier. Thus, there were $C_4^2 = 4 \times (4-1)/2 = 6$ classifiers trained to distinguish four states, which were μ_{12}/μ_{21}(neutral and onset), μ_{13}/μ_{31}(neutral and apex), μ_{14}/μ_{41}(neutral and offset), μ_{23}/μ_{32}(onset and apex), μ_{24}/μ_{42}(onset and offset) and μ_{34}/μ_{43}(apex and offset). Each pairwise SVM was fully trained on frames belonging to the two classes it is supposed to classify. At each iteration of the testing algorithm, all the SVM classifiers were applied on every frame in the testing sequence. Theoretically, the output of the SVM is the distance between a test pattern and the hyperplane defined by the support vectors. In [20], Platt proposed a method to estimate posterior probability by fitting this output with a sigmoid function. It is shown in Equation (1), in which $h(x)$ is the distance between the testing data x with the decision boundary of the SVM and parameters A and B are estimated by maximum likelihood from a training set. Consequently, those SVMs produced predicted labels for each frame, along with confidence levels of these labels.

$$\begin{aligned} p(y=+1|x) &= g(h(x), w^T, b) \\ &= \frac{1}{1+exp(w^T h(x)+b)} \end{aligned} \tag{1}$$

C. Temporal Modelling

HMMs are well-known robust machine learning tools and have been applied successfully in speech recognition and analysis of facial expression dynamics. Besides emission probability (B), as previously mentioned, there are two other parameters which are used to define an HMM: transition probability (Λ) and initial probability (Π). Where Λ is the probability of different transitions between underlying states of the model, B is the probability of one observation belonging to each state and Π is the probability distribution of the initial frame in each image sequence. Among them, Λ and Π were estimated from the distribution of training data directly while B was trained on the emission output from the SVM classifiers, which is formulated by Platt in [20]. The transformation from the pairwise probability to the posterior probability is shown in Equation (2), where S indicates the number of state, μ_{ij} indicates the pairwise probability, i and j indicate the state indexes of each pair.

$$p(q_i|X) = \frac{1}{\sum_{j=1, j\neq i}^{S} \frac{1}{\mu_{ij}} - (S-2)} \tag{2}$$

Afterwards, using the Bayes theorem we can obtain the emission probability $p(X|q)$ from the posterior probability $p(q|X)$ by dividing the distribution of each state $p(q)$ (equation (3)).

$$p(X|q) = \frac{p(q|X)}{p(q)} \tag{3}$$

We modelled two sequences in our system: blink sequence which contains blinks and non-blink sequence which does not contain any blinks. In modelling non-blink sequence, only one of the four segments was used: the neutral state. Hence, there was only one kind of transition in this model: neutral to itself, which is shown in Fig. 1. Meanwhile, we modelled another sequence which represents a complete blink using four different temporal states: neutral, onset, apex and offset. The general form of this blink model is shown in the Fig. 2. The blink model allows transitions from every state to its next state, as well as, to itself (besides neutral), but also from offset back to neutral. We assumed that the blinking started from neutral, progressed through the rest of the states and finally returned to the neutral state. Since the only state transition in the non-blink model is from neutral to neutral, we avoided having the same transition in the blink model in order to better discriminate between the two models. Therefore, we kept only the first and the last frame as neural state during the blink sequence pre-segmentation so that there was no transition from neutral to itself in blink sequences.

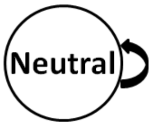

Fig. 1: Transitions of Non-blink Model

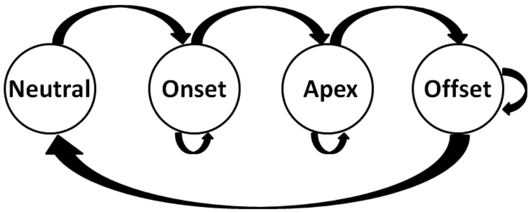

Fig. 2: Transitions of Blink Model

D. Real-time Processing Using Sliding Window

In order to detect blinks in real-time video streams (instead of pre-segmented sequences), we exploited a sliding window on the testing image sequence. The principle of the sliding window is explained in Fig. 3 below: It starts sliding from the first frame of the raw sequence from time T and after N steps it stops when the window reaches the end of the raw sequence.

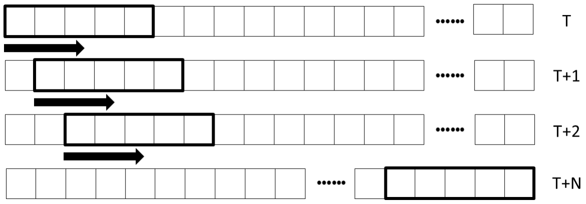

Fig. 3: Real-time Processing using Sliding Window

The sliding window allowed segmentation of the testing sequences to obtain much shorter sequences for processing, which saved on computation in each iteration. Once the segment was extracted, we evaluated its probability of whether contains a blink or not. In addition, we were able to decode each frame so that we obtained the frame predicted labels and could then estimate the blink duration by calculating the number of apex states (labelled as 3). In our system, the forward algorithm and Viterbi algorithm were used to solve the evaluation and decoding problems respectively.

Once the window began to slide and returned the frames within it, we evaluated the segment using the two pre-trained hybrid models, namely the blink model and non-blink model. The likelihoods, which describe how probable it is that this segment was generated by each model, were compared in order to determine the existence of a blink or not. The segment was predicted to contain blink if the blink model's likelihood was higher and vice versa.

Correspondingly, for every segment, we also decoded each frame using the model whose likelihood was higher during the comparison. However, some frames might be decoded more than once because they were captured in multiple sliding windows. For these frames, we applied majority voting in order to determine the final predicted label of each frame.

III. Experiment and Discussion

For the presented work, we conducted experiments using the gaze data recordings from the MAHNOB-HCI Database [21]. This data consists of spontaneous audio-visual data which records interactions between a participant and a computer. The recordings were made in a lab setting, using six cameras(61 fps), a uniform background and constant lighting conditions. There are three scenarios in the dataset: i) Office scenarios ii) Multimedia scenarios iii) Interaction with an avatar interrogator where the subject may lie or tell the truth in the conversation. We used the front-view camera and the third scenario in our experiments. The experiment uses the data recorded from 12 subjects (out of 33 in the dataset). For each subject, blink sequences varied much more significantly than non-blink sequence. Hence, we pre-segmented 40 video clips with blinks and 4 video clips without blinks for each subject. Additionally, in order to apply on-line testing, a 30 second video clip was segmented randomly (excluding the 44 clips) for each subject as well. DIKT [22], an on-line tracker, was applied accompanied with EyeAPI, an eye centre localization tool [23], to automate the extraction of eye region for each frame. Fig. 4 shows an entire blink sequence. Every frame of the dataset was annotated as neutral, onset, apex and offset state.

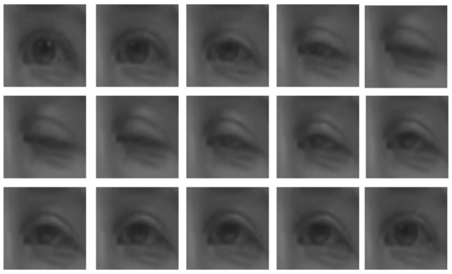

Fig. 4: Full Blinking Behaviour

Once We extracted features frame by frame for all sets of image sequences using methods described in Section 2.1. Those features were fed into classifiers and six pairwise SVMs were trained. We then fully trained two hybrid models, the blink and non-blink model, based on these SVMs through estimating three parameters, as described in Section 2.3. While the transformation from pairwise probability to posterior probability was implemented by Libsvm[24].

The layout of the following three sections are: 3.1) experiment adopting leave one subject out cross validation and classifying pre-segmented sequences as blink sequences or non-blink sequences. 3.2) experiment using leave one subject out cross validation and decoding each frame as one of the temporal states. 3.3) experiment exploiting a sliding window.

A. *Pre-segmented Sequences Classification*

In each iteration of this experiment, we left one subject out (40 blink sequences and 4 non-blink sequences) for test and trained on the other 11 subjects. Each pre-trained hybrid model, the blink and non-blink, applied on every test sequence and estimated a likelihood. We classified each sequence through comparing their likelihoods, which was described in Section 2.4.

We conducted testing using five different features and compared the results which are shown in Table I. In this table, we display the precisions, recalls and F1 measures for the five feature sets employed. After comparing, the intensity feature was found to discriminate blink segments from non-blink segments best.

	HOG	**INT**	**Gabor**	**LBP**	**OF**
Precision	94.58%	**99.38%**	98.75%	68.13%	97.50%
Recall	**100.00%**	99.58%	**100.00%**	**100.00%**	**100.00%**
F1 measure	97.22%	**99.48%**	99.38%	81.04%	98.74%

TABLE I: Classification Results of Per-segmented Sequences(INT means pixel intensity, Gabor means Gabor filter and OF means Optical Flow)

B. *Frame By Frame Classification*

There is only one state in the non-blink model so that this model always classifies every frame as neutral. Thus, we conducted this experiment only using blink model and blink sequences. In each iteration, we left one subject out (40 blink sequences) for test and trained on the other 11 subjects. The pre-trained blink model classified each frame of the testing sequences as one of the temporal states: neutral, onset, apex and offset. In order to compare with the temporal model, we employed a four-class SVM using Libsvm [24] as a classifier without temporal information.

Table II displays the classification accuracy of two different approaches and five different features exploited. After introducing temporal information, the accuracy was found to increase significantly when using all features with optical flow displays the best discriminative ability. However, HOG and Gabor filter were found to struggle from distinguishing the onset from the offset since eyes appear narrow in both states and both these features are edge detection descriptors. By contrast, optical flow can classify these two states easily by calculating eye motion direction. However, sometimes optical flow failed to differentiate neutral and apex states as the eyes rarely move in these cases.

C. *Testing With The Sliding Window*

In this experiment, a sliding window was applied on the testing sequence in order to detect blinks and calculate blink durations in real-time. In each iteration, we left one subject (randomly segmented 30 second video clip) out for test and trained two models on the pre-segmented sequences (40 blink sequences and 4 non-blink sequences) of the other 11 subjects. During testing, for each segment extracted by the window, the two pre-trained hybrid models evaluated and classified it as a sequence with or without blinks. The blink model was exploited to decode each frame if the segment was detected to contain blink. Otherwise, we used non-blink model to decode each frame as neutral state. After applying majority voting on those frames which were decoded for many times, we calculated the number of apex states and estimated blink durations.

	HOG	**INT**	**Gabor**	**LBP**	**OF**
4-class SVM	52.45%	50.19%	54.45%	48.00%	**62.83%**
Hybrid Model	78.9%	69.96%	72.99%	66.06%	**79.36%**

TABLE II: Frame-by-frame Classification Results

Among all the testing results, 90.99% of the blinks were recognized successfully while 7.21% of the patterns were misclassified as blinks. Even all estimations of blink durations were longer than the ground truth, we could still spot spontaneous blinks from voluntary blinks. A real-time detection result for one of the subjects is shown in Fig. 5.

IV. Conclusion

In this paper, we have demonstrated a hybrid system combining HMM and SVM for automatic eye blink detection and blink duration calculation. Several popular blink detection features were extracted and their performances were compared. In particular, pixel intensity has the best performance in sequence classification while optical flow achieves highest accuracy in decoding per frame. Unlike previous work, we modelled blink temporal dynamics into our system. As a result, the temporal model works significantly better than multi-class SVM when classifying each frame. However, the frame-by-frame classification accuracy is still not good enough and one of the reason might be the ambiguity in annotation: sometimes the difference of eyes between each state is not obvious.

References

[1] M. Knapp, *Lying and deception in human interaction.* Allyn and Bacon, 2008.

[2] M. Hartwig and C. Bond Jr, "Why do lie-catchers fail? a lens model meta-analysis of human lie judgments." *Psychological bulletin*, vol. 137, no. 4, p. 643, 2011.

[3] J. Burgoon, "Nonverbal measurement of deceit," *The sourcebook of nonverbal measures: Going beyond words*, pp. 237–250, 2005.

[4] K. Fukuda, "Eye blinks: new indices for the detection of deception," *International Journal of Psychophysiology*, vol. 40, no. 3, pp. 239–245, 2001.

[5] Y. Tian, T. Kanade, and J. Cohn, "Dual-state parametric eye tracking," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on.* IEEE, 2000, pp. 110–115.

[6] ——, "Eye-state action unit detection by gabor wavelets," *Advances in Multimodal InterfacesICMI 2000*, pp. 143–150, 2000.

[7] K. Grauman, M. Betke, J. Gips, and G. Bradski, "Communication via eye blinks-detection and duration analysis in real time," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–1010.

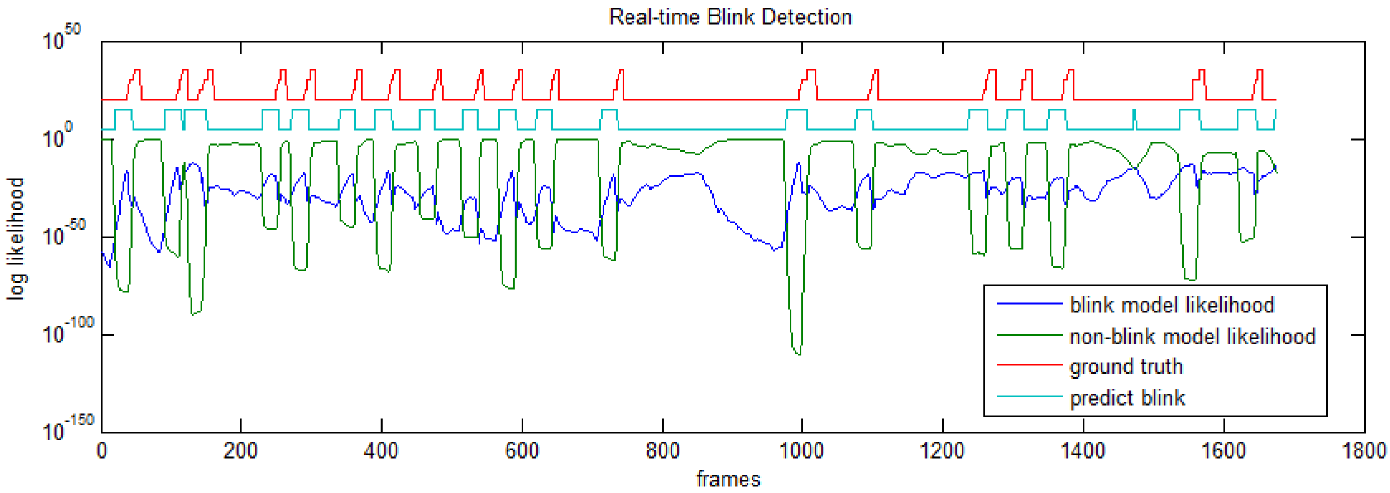

Fig. 5: Real-time Blink Detection

[8] M. Khan and A. Mansoor, "Real time eyes tracking and classification for driver fatigue detection," *Image Analysis and Recognition*, pp. 729–738, 2008.

[9] T. Ito, S. Mita, K. Kozuka, T. Nakano, and S. Yamamoto, "Driver blink measurement by the motion picture processing and its application to drowsiness detection," in *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*. IEEE, 2002, pp. 168–173.

[10] T. Bhaskar, F. Keat, S. Ranganath, and Y. Venkatesh, "Blink detection and eye tracking for eye localization," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, vol. 2. IEEE, 2003, pp. 821–824.

[11] R. Heishman and Z. Duric, "Using image flow to detect eye blinks in color videos," in *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*. IEEE, 2007, pp. 52–52.

[12] A. Krolak and P. Strumillo, "Vision-based eye blink monitoring system for human-computer interfacing," in *Human System Interactions, 2008 Conference on*. IEEE, 2008, pp. 994–998.

[13] J. Li, "Eye blink detection based on multiple gabor response waves," in *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 5. IEEE, 2008, pp. 2852–2856.

[14] T. Moriyama, T. Kanade, J. Xiao, and J. Cohn, "Meticulously detailed eye region model and its application to analysis of facial images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 5, pp. 738–752, 2006.

[15] K. Minkov, S. Zafeiriou, and M. Pantic, "A comparison of different features for automatic eye blinking detection with an application to analysis of deceptive behavior," in *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*. IEEE, 2012, pp. 1–4.

[16] M. Pantic and I. Patras, "Detecting facial actions and their temporal segments in nearly frontal-view face image sequences," in *Proceedings of IEEE Int'l Conf. Systems, Man and Cybernetics (SMC'05)*, Waikoloa, Hawaii, October 2005, pp. 3358–3363.

[17] S. Koelstra and M. Pantic, "Non-rigid registration using free-form deformations for recognition of facial actions and their temporal dynamics," in *Proceedings of IEEE Int'l Conf. Automatic Face and Gesture Recognition (FG'08), Amsterdam, Netherlands*, September 2008, pp. 1–8.

[18] O. Rudovic, V. Pavlovic, and M. Pantic, "Kernel conditional ordinal random fields for temporal segmentation of facial action units," in *Proceedings of the 12th European Conference on Computer Vision (ECCV-W'12). Florence, Italy*, October 2012.

[19] M. F. Valstar and M. Pantic, "Fully automatic recognition of the temporal phases of facial actions," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 42, pp. 28–43, 2012.

[20] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[21] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multi-modal database for affect recognition and implicit tagging," *IEEE Transactions on Affective Computing*, p. 1, July 2011, in press.

[22] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic, "Fast and robust appearance-based tracking," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG'11)*, Santa Barbara, CA, USA, March 2011, pp. 507–513.

[23] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[24] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.