A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection: Supplementary Material

Reza Ghoddoosian Marnim Galib Vassilis Athitsos
Vision-Learning-Mining Lab, University of Texas at Arlington
{reza.ghoddoosian, marnim.galib}@mavs.uta.edu, athitsos@uta.edu

1. Blink Retrieval Algorithm

In our experiments, we noticed that the approach of Soukupová and Cech [3] typically detected consecutive quick blinks as a single blink. This created a problem for subsequent steps of drowsiness detection, since multiple consecutive blinks can typically be a sign of drowsiness. We added a post-processing step on top of the output of [3], that successfully identifies the multiple blinks which may be present in a single detection produced by [3].

According to[3], define EAR, for each frame, as below:

$$EAR = \frac{||\vec{p_2} - \vec{p_6}|| + ||\vec{p_3} - \vec{p_5}||}{||\vec{p_1} - \vec{p_4}||} \tag{1}$$

In the above, each $\vec{p_i} \in \{p_i | i = 1, ..., 6\}$ is the 2D location of a facial landmark from the eye region, as illustrated by Figure 1. In [3], an SVM classifier detects eye blinks as a pattern of EAR values in a short temporal window of size 13 depicted in Fig.2. This fixed window size is chosen based on the rationale that each blink is about 13 frames long. A single blink takes around 200ms to 400ms on average [1, 2], which translates to six to twelve frames for a video recorded at 30fps. Even if 13 frames is a good estimate for the length of a blink, this approach would not handle consecutive quick blinks.

As depicted in Figure 2, each value in this 13 dimensional vector corresponds to the EAR of a frame with the frame of interest located in the middle. The SVM classifier takes these 13D vectors as input and classifies them as "open" or "closed" (more specifically referred to the frame of interest in each input vector). A number of consecutive "closed" labels represent a blink with the length of M. Subsequently, the EAR values of these M frames are stored in $\mathbf{x}$ in order, and fed to the "Blink Retrieval Algorithm", explained in Alg.1, for post-processing (Fig. 3a). The sequence of EAR values for **one** blink by [3] will be considered as a **candidate** for **one or more than one** blinks.

This algorithm runs in $\Theta(M)$ time, where M is the number of frames in the video segment that is used as input to the algorithm. In practice, the algorithm runs in real time. In

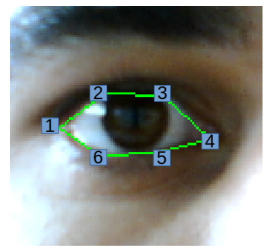

Figure 1. Six points marking each eye.

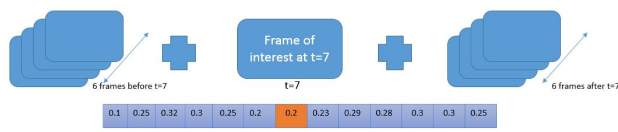

Figure 2. Presenting each frame (at t=7) by 13 numbers (EARs) concatenated from 13 frames as a feature vector.

addition, Alg.1 sets a definite frame on when a blink starts, ends or reaches its bottom point based on the extrema of its EAR signal. For better results, x is passed through a median/mean filter to clear the noise and then fed to the algorithm.

At step 1, the derivative of $\mathbf{x}$ is taken. Then, zero derivatives are modified, at steps 2 and 3, so that those derivatives have the same sign as the derivative at their previous time step. This modification helps to find local extrema, as points where the derivative sign changes (steps 4 to 7). The threshold, defined at step 8, is used to suppress the subtle ups and downs in $\mathbf{x}$ due to noise and not blinks. The extrema in $\mathbf{x}$ are circled in Figure 3b, and labeled (+1 or -1) relative to the threshold (steps 9 to 11). Each two consecutive extrema are indicative of a downward or upward movement of eyes in a blink if those two are connected, so that the link or links between them pass the threshold line (steps 12 and 13). Fig.3c highlights these links in red. Finally, each pairing of these red links corresponds to one blink with start, end and bottom points as depicted in Figure 3d (steps 14 to the end).

Algorithm 1 Blink Retrieval Algorithm

Input The initial detected EAR signal $\mathbf{x} \in \mathbb{R}^M$, where M is the size of the $\mathbf{x}$ time series, as a candidate for one or more blinks and epsilon=0.01

Output N retrieved blinks, $N \ll M$

1: $\dot{\mathbf{x}}[n] \leftarrow \mathbf{x}[n+1] - \mathbf{x}[n], \forall\, n \in \{i|i = 0, 1, ..., M-2\}$
2: **if** $\dot{\mathbf{x}}[0] = 0$ **then** $\dot{\mathbf{x}}[0] \leftarrow -1 \times$ epsilon
3: $\dot{\mathbf{x}}[n] \leftarrow \dot{\mathbf{x}}[n-1] \times$ epsilon, $\forall\, n \in \{i|\dot{\mathbf{x}}[i] = 0 \wedge i \neq 0\}$ to avoid zero derivatives for steps 4 and 6
4: $\mathbf{c}[n] \leftarrow \dot{\mathbf{x}}[n+1] \times \dot{\mathbf{x}}[n], \forall\, n \in \{i|i = 0, 1, ..., M-3\}$
5: Define $\mathbf{e} \in \mathbb{R}^{P+2}, P \leq M-2$ to store the indices for the P extrema, the first and the last points in $\mathbf{x}$
6: $\mathbf{e}[0] \leftarrow 0$, $\mathbf{e}[P+1] \leftarrow M-1$, supposing the first and last points in $\mathbf{x}$ are maxima
7: $\mathbf{e}[k] \leftarrow n+1, \forall\, (n \in \{i|\mathbf{c}[i] < 0\} \wedge k \in \{i|i = 1, 2, ..., P\})$ ▷ Indices of P+2 extrema, including the first and last points in $\mathbf{x}$ are stored in order
8: Define $THR \leftarrow 0.6 \times \max(\mathbf{x}) + 0.4 \times \min(\mathbf{x})$, as a threshold
9: Define $\mathbf{t} \in \mathbb{R}^{P+2}$, to store +1 or -1 for extrema above and below threshold respectively
10: $\mathbf{t}[0] \leftarrow +1, \mathbf{t}[P+1] \leftarrow +1$, supposing the first and last points in $\mathbf{x}$ are maxima
11: Append +1 in $\mathbf{t}$ for each $n \in \{i|\mathbf{x}[\mathbf{e}[i]] > THR\}$, and append -1 in $\mathbf{t}$ for each $n \in \{i|\mathbf{x}[\mathbf{e}[i]] \leq THR\}$, all in the order of the indices in $\mathbf{e}$
12: Define $\mathbf{z} \in \mathbb{R}^{P+1}$, $\mathbf{z}[n] \leftarrow \mathbf{t}[n+1] \times \mathbf{t}[n]$
13: Define $\mathbf{s}$, to store the indices of all negative values in $\mathbf{z}$, representing the downward and upward movements of eyes in a blink
14: $N \leftarrow \frac{length(\mathbf{s})}{2}$ ▷ N is the number of sub blinks, and $length(\mathbf{s})$ is always an even number
15: **for** i ←0 to $N-1$ **do** ▷ Define for $blink_i$:
16: StartIndex $\leftarrow \mathbf{e}[\mathbf{s}[2 \times i]]$,
17: EndIndex$\leftarrow \mathbf{e}[\mathbf{s}[2 \times i + 1] + 1]$,
18: BottomIndex$\leftarrow \mathbf{e}[\mathbf{s}[2 \times i + 1]]$

return start, end and bottom points of the N retrieved blinks in $\mathbf{x}$

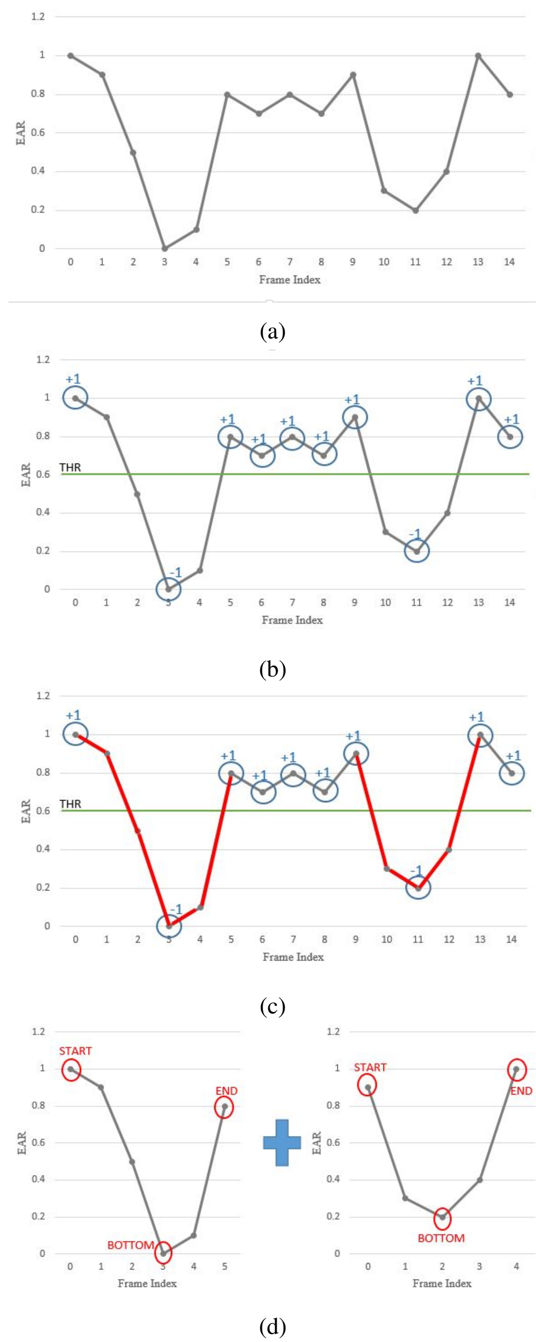

Figure 3. The Blink Retrieval Algorithm steps: (a) $\mathbf{x}$ with size $M = 15$ as the input for Alg. 1. (b) The indices of circled points form $\mathbf{e}$, and the set of +1 and -1 labels forms $\mathbf{t}$ with $P = 8$. (c) The red lines indicate where $\mathbf{z}$ values are negative. (d) Two ($N = 2$) blinks are retrieved with definite start, end and bottom points.

References

[1] S. ROSTAMINIA, A. MAYBERRY, D. GANESAN, B. MARLIN, and J. GUMMESON. ilid: Low-power sensing of fatigue and drowsiness measures on a computational eyeglass. 2017. 1

[2] M. Singh and G. Kaur. Drowsy detection on eye blink duration using algorithm. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):363–365, 2012. 1

[3] T. Soukupova and J. Cech. Real-time eye blink detection using facial landmarks. In *21st Computer Vision Winter Workshop (CVWW2016)*, pages 1–8, 2016. 1