A REVIEW OF VIDEO-BASED AND MACHINE LEARNING APPROACHES TO HUMAN EYE BLINK DETECTION IN VIDEO

by
DYANA MULLER

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors Degree with Honors in
Computer Science

THE UNIVERSITY OF ARIZONA

MAY 2019

Approved by:

Dr. Kobus Barnard
Department of Computer Science

Abstract

Automation of detection of human eye blink in video has a broad array of applications, including detection of disease, anti-spoofing software, and helping individuals with physical disabilities interact with computers. The present work provides a review of several papers within the past two decades which propose methods for automated blink detection, highlighting the evolution of the field alongside developments in machine learning techniques. Then, the strengths and shortcomings of several popular approaches are evaluated in the context of eye blink detection. Namely, I focus on appearance-based and motion-based computation, support vector machines, convolutional neural networks, and long short-term memory networks. Finally, I report the beginnings of a reproduction of the methods outlined in one of the papers reviewed.

A Review of Video-Based and Machine Learning Approaches to Human Eye Blink Detection in Video

Introduction

Eye blink detection and analysis has numerous applications in fields ranging from medicine to technology to security. In the medical field, spontaneous eye blink has been consistently correlated with dopamine levels in certain areas in the brain (Taylor et al. 1999), making it a popular and reliable proxy for measuring dopaminergic function (Jongkees and Colzato 2016). Several conditions related to abnormal dopaminergic function have been associated with some deviation from normal blink behavior, for example, increased blink rate in patients with schizophrenia (Chen et al. 1996; Karson 1983; Freed et al. 1980), decreased blink rate in those with Parkinson's disease (Deuschl 1998; Karson 1983; Karson et al. 1982), and decreased blink duration in depressed subjects (Alghowinem et al. 2013). Blink rate analysis may therefore offer an inexpensive and noninvasive way of detecting certain medical conditions. Eye blink tracking is also a key component of driver assistance systems, which monitor driver vigilance levels (Du et al. 2008). In the security field, eye blink detection can used to improve the reliability of face recognition authentication software. Without the use of blink detection, these can often be fooled by moving a photograph of a face in front of the camera (Pan et al. 2007). Beyond these examples, there is potential use for blink detection in helping disabled people interact with computers (Grauman et al. 2003), assess cognitive effort (Wilson 2002), identify lies (Fukuda 2001), or create natural animations (Trutoiu et al. 2011).

The applications of eye blink detection are robust and far-ranging, but all of them require or could highly benefit from automated detection from video. While electromyography (EMG), which captures signals from muscle movements around the eye, is a quick and relatively accurate way to capture blink information, it requires sensors to be affixed to the target at all times during blink detection. This may not be a practical option for many of the previously-mentioned applications.

Therefore, there is significant demand for fast, accurate, non-invasive methods of blink detection. Within the past decade, several groups in computer vision have tackled the challenge of developing methods to accurately identify the occurrence of blinks in video. This literature review will begin with an overview of several papers which outline different methods for automatic blink detection in video; then, some of the approaches will be evaluated in greater detail.

Overview of Blink Detection Papers

Most successful blink detection techniques began in the early 2000s. During the earlier years, the bulk of techniques were purely computational, extracting specific features from an image, using them to compute some output value, and comparing that value to some bound or threshold to determine whether a blink occurred or not. These methods fit into two categories: appearance-based and motion based. Appearance-based algorithms use features related to the content of each pixel, while motion-based ones are interested in how a particular point on the face moves from one frame to the next.

One such appearance-based approach is that used by Marcos-Ramiro et al. (2014). They employ an off-the-shelf face tracker to localize the eyes. Then, within the eye region, they apply a Random Forest Classifier to associate each pixel with one of three eye parts: pupil/iris, sclera, or eyelid. For each frame, they calculate eye openness, defined as the number of pixels classified as pupil/iris or sclera as compared to eyelid, as well as the difference in eye openness compared to the two previous frames. This value is compared to a threshold to determine if a blink occurred. They achieve a 93.65% F1 score, which was outperformed by previous studies, but the approach has the benefit of providing similar accuracy despite face rotations, distance to camera, and high appearance variance across subjects.

In 2014, Drutarovsky et al. proposed a motion-based algorithm based on motion vectors. First, trackers are placed in several positions in a grid formation over the eye region. The local motion vectors at these points are extracted and the horizontal components isolated. The variance of the vertical components acts as the input for the state machine. High variance in the middle portion of the defined eye region as compared with the outer portions is indicative of a blink. They evaluate their method with a novel dataset which they call Eyeblink8, consisting of 8 videos from 4 individuals. The research group reports 99.8% accuracy on both the Talking Face and ZJU datasets; however, because this approach relies on the vertical component of motion vectors, it is very vulnerable to tilted faces.

Grauman et al. (2001) proposed a combination of appearance-based and motion-based analysis. First, the sequence of video frames is searched for “blink-like motion”. This is done by computing a difference image from one frame to the next to obtain possible eye candidates. Then, the motion components of pairs of these candidates are compared to known blink measurements to determine which pair is most likely to be the eyes. Once the eyes have been located, they are compared over time to templates of an open and a closed eye. Concurrent dips in the correlation score with an open eye and spikes in the correlation score with a closed eye indicate a blink. With this method, Grauman et al. achieved an accuracy of 95.6%.

In 2009, the creation of ImageNet launched an artificial intelligence boom. ImageNet is a database of more than 14 million hand-annotated images, an incredible tool for the computer vision community. With this boom came the popularization of many machine learning techniques to solve existing problems. One of these was the Support Vector Machine (SVM), a type of supervised learning classifier. When applied to blink detection, is trained on various features of the input video frames to classify an eye as open or closed. One classic training feature is the ratio of the height to the width of the eye. Lee et al. (2010) uses this ratio, first beginning with a simple face tracker coupled with a single SVM, and then working to improve its performance. In their first iteration, they use the AdaBoost face detector to locate the eyes, then use a single SVM trained on the previously mentioned height-to-width ratio. They achieve 64.62% recall and 88.1% precision. To improve these results, they couple the AdaBoost face detector with the Lucas-Kanade-Tomasi (LKT) feature tracker to increase robustness in the case of face rotation. With this change, they achieve 84.55% recall and 92.41% precision. As a final improvement, they use an additional SVM trained on the number of black pixels in the eye region, which increases as the eye closes and covers up the white eye sclera. When tested across seven different datasets, including ZJU, Talking Face, and four of their own, their final average recall was 92.3% and their average precision 94.67%. However, one significant pitfall is the lack of ability to incorporate temporal information, meaning that it incorrectly detected blink when the subject was looking down, even if they were doing so for an extended period of time.

Sun et al. had the same observation: that previous methods classified the eye based only on the information provided in a single frame. In 2013, they proposed an improvement which incorporated temporal information using a combination of Hidden Markov Models (HMMs) and SVMs. They trained six multi-class SVMs to distinguish between all possible pairs of four different states: neutral (open eye), onset (closing eye), apex (closed eye), and offset (opening eye). With these six classifiers, they trained two models, one for blink and one for non-blink. The non-blink model allowed only for a transition from the neutral state to itself, while the blink model allowed transitions from neutral to onset to apex to offset back to neutral, and from each of these states to themselves. Each of these models is illustrated below.

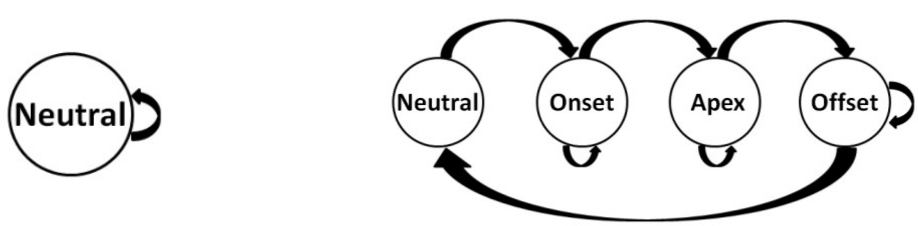

Figure 1. Transitions of a non-blink model (left) and blink model (right) (Sun et al. 2013).

For detection of blink, a sliding window of the video frames in the sequence is passed into the two models. Whether a blink has occurred or not during the sequence of frames assessed is determined by the model which returns the highest probability. They compared their performance when training their models on five different image features: histogram of oriented gradients, Gabor filter, local binary pattern, optical flow, and pixel intensity. Their highest F1 score (combined measure of recall and precision) of 99.48% was achieved when training on pixel intensity. However, regardless of training feature, their model was not able to correctly classify more than 79.36% of individual frames, injecting some doubt about the accuracy of their model.

All the papers discussed so far follow the same general pattern. First, some type of face or eye detector is used to identify the eye region; then, features from this region are used to determine if a blink occurred. Wu et al. (2010) were unsatisfied with this observation, noting that this approach relies on the face detector being able to robustly handle the appearance change of an open to a closed eye. Therefore, they sought to develop a method which simultaneously tracks eyes and detects blink. A bit off the beaten path, their computation involves a binary tree-based statistical structure of the features which encodes dependency between features. First, a binary classifier searches the frame exhaustively for eye patches. Once eyes are found, three types of information are extracted from the sequence: global motion of eyes, local motion of pupils, and eye openness or closure. Conditionally-independent features are separated into different subtrees, while more dependent features are kept in the same node. If two subtrees share the same parent, then they are statistically very close. The result is a tree which at any given level represents the relevant features for the object at the corresponding scale. The model is trained on annotated data to obtain a tree which effectively models the feature set. Using this tree, blink can be determined by recursively calculating the joint probability for features in a feature clique. The authors claim their algorithm accurately tracks eye locations, detects both voluntary and involuntary blinks, and accurately records blink duration. However, they only provide a true positive and false positive count for a few different proprietary video sequences; they lack a recall, precision, or accuracy measurement and do not evaluate their model on known datasets. Additionally, the implementation appears to be quite computationally complex with a disproportionately low amount of payout. A recurrent neural net with a reliable eye tracker, which is not hard to come by these days, would achieve similar, if not better, results.

As machine learning has progressed, the idea of deep learning has gained much popularity in the past few years. Deep learning, as opposed to shallow learning, uses a network containing multiple layers. Because of the additional layers, deep learning methods tend to be more accurate. Among the first to use a CNN for classification of open and closed eyes was Kim et al. (2017). Their CNN is based on the ResNet-50, a 50-layer residual network. This differs from a normal CNN in that instead of learning features, it learns residual, i.e. the difference between the feature learned and the current layer's input. In 20 epochs of training, this method was able to outcompete other CNNs, namely VGG, GoogLeNet, and AlexNet, in terms of number of false negatives and false positives when classifying images as open or closed eye. In the same year, Anas et al. went a step further and used a CNN to detect blink. They trained their model on the Helen dataset, which consists of facial images representing subjects of different age, gender, and ethnic origin, and which have variable illumination, resolution, and face position. Their CNN learned to discriminate between three classes: open, partially-open, and closed, and they found that it outperformed CNNs like the one in Kim et al., which only discriminate between open and closed eye.

One downside of large CNNs is the resources they require to train the model. Han et al. (2018) proposed a hybrid SVM-CNN approach which could be used on small devices such as smartphones. They first used an SVM classifier trained on histogram of oriented gradients features to identify the eye region. This region is then passed to a LeNet-5 CNN model, a small, 5-layer CNN, to distinguish between open eye, closed eye, and background. Training is done with a dataset that consists of image clips of open and closed eyes, as well as random image clips for the background class. They evaluate their model using the ZJU dataset, and unsurprisingly, they report their model to perform much better than a multi-class SVM. Additionally, they report that the software is able to run without problems on a smartphone.

Several types of neural networks can also be combined to achieve better performance. Li et al. (2018) combine a CNN with a recurrent neural network (RNN), resulting in a long-term recurrent convolutional neural network (LRCN). First, they detect the face in each frame and extract the eye region using a facial landmark detector. The CNN of the LRCN first identifies discriminative features. The output is passed to an RNN with long short-term memory (LSTM) cells for sequence learning. Finally, the output is classified as corresponding to either open or closed. They train their model on a proprietary dataset of 50 30-second videos containing one individual each with at least one blink occurring. Compared to a CNN-only approach, the authors report the LRCN model performs better due to the ability to factor in the states from the images immediately preceding it. The intuition is that if blinking has just occurred, the eye is likely to be open in the next few frames.

With these examples which range in publish date from 2001 to 2018, we can see how the eyeblink detection field has evolved over the past couple decades. In the earlier years, most methods used a computational approach based on the content of each pixel or the movement of a facial feature from one frame to the next. With the popularization of machine learning techniques came the emergence of approaches such as SVMs and HMMs. Then, with the introduction of modules and packages for deep learning such as Keras, Caffe, and TensorFlow, among many others, implementation of deep learning models became much more accessible. This allowed researchers to easily experiment with modified networks and even combinations of several types of networks. Today, we continue to see improved performance as new machine learning techniques are integrated into eye blink detection.

Having seen an overview of several different types of approaches, we will now examine each type in more detail, evaluating their features, strengths, and shortcomings.

Evaluation of Blink Detection Methods

Almost all blink detection approaches follow the same pattern: first, the eye region is identified; then, a blink detection method is applied to that region. In earlier years, common methods for eye detection included gradient flow fields, color-based method for detection of eye sclera, horizontal gradient maps of skin-colored region, and pupil detection using infrared lighting (Grauman 2001). However, now that face recognition software has advanced to a point where it can reliably detect faces and accurately label facial features, in the past few years there has not been much variation in how different groups approach identifying the eye region. Most use an off-the-shelf method such as AdaBoost, Viola Jones, or Lucas-Kanade-Tomasi (LKT). There is, however, significant variation in how blinks are detected once the region of interest has been restricted to the eyes. The methods which were most popular during the initial phase of the eye blink detection era were video-based computational methods. These can then be split into the categories of appearance-based and motion-based. Some examples of common appearance-based features include ratio of eye height to width, pixel intensity, histogram of gradients (HOG), and optical flow. The pitfall of appearance-based methods is that they tend to be very inaccurate in the face of variability in appearance, such as presence of glasses, prominent eyebrows, small natural eye-openness, dark skin color, etc. Motion-based methods, on the other hand, are unaffected by such appearance changes, as they only track the movement of points on the face from one frame to the next. Where motion-based methods fall short is a tendency for a high false positive rate due to vertical head movements which are misinterpreted as vertical eyelid movements.

Machine learning captures the vast majority of non-video-based approaches. These techniques involve training a model on a dataset so that it “learns” what features contribute to a blink occurring or not. When given examples it has not seen before, the model will ideally be able predict the correct outcome using the features it has learned. In general, these approaches tend to be more accurate and, after having been trained, require less computational processing than appearance-based or motion-based computational approaches. Machine learning techniques can be broken down into deep learning and shallow learning categories, where deep learning encompasses neural networks with more than one internal layer and shallow learning is anything which is not deep learning. There is a vast selection of both shallow and deep learning techniques currently on the market, but for the purposes of this paper, I will only examine SVMs, CNNs, and LSTMs.

Support vector machines (SVMs) constitute one of the beginnings of machine learning, and are an example of shallow learning. Given a training set of data labeled as belonging to one of two categories, these points are plotted into a feature space, where the feature dimensions are previously defined. The SVM then learns the model of a hyperplane which best separates the points belonging to each category. The optimal hyperplane is that which correctly classifies the most number of points while also maintaining the greatest margin around the hyperplane (see Figure 3 below). The hyperplane may pass through as many or as few feature dimensions as necessary to achieve this goal.

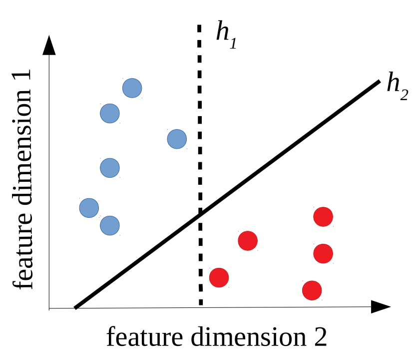

Figure 3. SVM selection of optimal hyperplane. While hyperplane h_1 separates the data with no misclassifications, the margin around the hyperplane is much less than that of h_2, so the SVM will select h_2.

As previously mentioned, SVMs tend to perform better and require less processing than non-machine learning approaches. However, SVMs only support binary classification. While they may perform great on a classification task between open and closed eyes, this may not be sufficient to determine whether a blink has occurred or not in a video sequence. Additionally, because the feature dimensions must be chosen in advance, we encounter a similar issue as with appearance-based feature extraction: we rely on our chosen features being able to optimally capture class differences. What's more, we must have some intuition beforehand about what features might contribute to class differences. In the context of blink detection where the two classes are open and closed eye, this may not be so much of an issue. However, we could surely benefit from a method which could automatically determine which features are most important.

Convolutional neural nets (CNNs) are a deep learning approach which overcome some of these issues. They are made up of an input layer, a sequence of hidden layers which perform feature learning, another sequence of hidden layers which perform the classification, and an output layer. The feature learning sequence contains convolutional and pooling layers which reduce the size of the input while maintaining features important for prediction. In this way, they are able to capture low-level features about the input image, such as edges, color, and gradient orientation. With an increased number of convolutional layers, they can also capture high-level features, producing a more complete understanding of the image. The classification layers allow for prediction based on non-linear combinations of the learned features. Unlike the previously mentioned methods, CNNs have the advantage of being able to learn which features are most relevant. This helps maximize the relevant feature space, maximizing the prediction power. Moreover, it is an especially desired trait when there is little intuition about the features which separate two classes. This characteristic is the reason why CNNs are much more resistant to subject motion, face tilt and rotation, and blurry images, all of which pose a significant challenge for all the previously-described methods. Besides optimized feature extraction, learning the feature space during training comes with another advantage: a relatively low amount of pre-processing is required. However, despite several advantages, CNNs still fail to address one of the major shortcomings of appearance-based, motion-based, and SVM techniques, which is the lack of encoding of temporal information.

Long short-term memory (LSTM) neural networks are another deep learning technique which, as their name might suggest, are able to incorporate temporally-related features due to an ability to "remember" previous states. The units that make up an LSTM contain an input gate, output gate, and a forget gate. Together, they control the amount of information that flows into and out of a cell. In other words, they determine how much information from the current input and how much from the previous states goes into predicting the current state. The ability to incorporate past predictions makes LSTMs ideal for sequences such as videos. In the context of eye blink detection, temporal information is useful because if a blink has just occurred, it is unlikely that a blink will occur in the next few frames.

In summary, machine learning approaches tend to be more accurate than other methods because of their ability to learn an optimal model for the classification problem. Among machine learning techniques, deep learning ones have several advantages over shallow learning ones. In particular, they are able to learn a more holistic representation of the input, giving it improved prediction power and a decreased need for image pre-processing. Deep neural networks which incorporate information from previous states into the current prediction are well-suited for temporal sequences. This makes LSTMs a powerful approach for eye blink detection. It is no coincidence that Li et al. (2018) achieve an improved performance as compared to other approaches using their combination of a CNN and LSTM. The main disadvantage of machine learning techniques is that they must be trained on a robust dataset. Depending on the complexity of the network, the process of assembling an adequate dataset and training the network could take weeks. However, barring insufficient time or a trivial classification problem, the benefits of using a deep learning approach outweigh the costs. Especially now that many easy-to-use deep learning modules are publicly available, implementing deep learning approaches is more rewarding than ever.

Beginnings of a Replication of Li et al. (2018)

The Interdisciplinary Visual Intelligence Lab (IVILAB) under Dr. Kobus Barnard at the University of Arizona is home to several very challenging eye blink datasets. The videos in these sets have low illumination and low resolution, resulting in poor predictive performance using several machine learning models. Noting Li et al. (2018)'s success with a combined CNN-LSTM, I sought to reproduce their model for use on these datasets. I was able to obtain the 32 raw videos the authors used for training, as well as pickle files annotating whether a blink is present in each frame of each video. To create a dataset from these videos, each frame needed to be extracted and the eye regions cropped from each frame. I wrote three bash scripts to achieve these endpoints. The first extracted every frame out of each video. The second applied the OpenFace facial landmark detector to each frame, outputting an annotation file containing the coordinates for 68 points on the face, including 6 points on each eye. The final script used this face annotation file to create a bounding box around the eye and crop each frame of each video to this bounding box. The result is a complete dataset which can be passed into any model for training. Additionally, I have used python's Keras module to create a CNN like the one described in Li et al. (2018). This model is based on the VGG16, a pre-trained 16-layer CNN. Instead of the VGG16's last fully connected layer 7 and 8, Li et al.'s CNN has three untrained fully connected layers tacked onto the end. To implement Li et al.'s model, all that is missing is to connect to the CNN to an LSTM and train the two jointly on the dataset I have created. Both the dataset and the untrained, modified VGG16 model is available on the IVILAB computer named bayes01 and is located at `/data/faces/blink/in_ictu_oculi.` They are open for use by anyone else in the IVILAB looking to continue replicating the model described in Li et al. (2018) or anyone who could make use of an annotated dataset of human eye blink.

Impact

The present work is a review of several different approaches to eye blink detection, tracking the evolution from video-based computational methods to shallow machine learning to deep machine learning. What follows is a broad evaluation of the strengths and weaknesses of these three categories of approaches for solving the eye blink detection problem. Finally, I present the progress I have made with replicating a deep learning model described in one of the reviewed papers. This review is a helpful introduction for someone looking to better understand the problem of eye blink detection, as well as the pros and cons of different machine learning approaches in general. Additionally, the dataset I have created may be a useful tool for anyone implementing a blink detection method, and the modified CNN is a starting point for anyone else looking to continue the replication of Li et al. (2018)'s model.

Works Cited

Alghowinem, S., R. Goecke, M. Wagner, G. Parker, and M. Breakspear (2013). Eye movement analysis for depression detection. *IEEE International Conference on Image Processing 2013.*

Anas, E.R., P. Henriquez, B.J. Matuszewski (2017). Online eye status detection in the wild with convolutional neural networks. *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP):* pp. 88-95.

Chen, E.Y.H., L.C.W. Lam, R.Y.L. Chen, and D.G.H. Nguyen (1996). Blink rate, neurocognitive impairments, and symptoms in schizophrenia. *Biological Psychiatry* 40.7: pp. 597-603.

Deuschl, G. and C. Goddemeier (1998). Spontaneous and reflex activity of facial mescles in dystonia, Parkinson's disease, and in normal subjects. *Journal of Neurology, Neurosurgery & Psychiatry* 64.3.

Drutarovsky, T. and A. Fogelton (2015). Eye blink detection using variance of motion vectors. *ECCV 2014 Workshops:* pp. 436-448.

Du, Y., P. Ma, X. Su, and Y. Zhang (2008). Driver fatigue detection based on eye state analysis. *Proceedings on the 11th Joint Conference on Information Sciences 2008*.

Freed, W.J., J.E. Kleinman, C.N. Karson, S.G. Potkin, D.L. Murphy, R.J. Wyatt (1980). Eye-blink rates and platelet monoamine oxidase activity in chronic schizophrenic patients. *Biological Psychiatry* 15.2: 329-332.

Fukuda, K. (2001). Eye blinks: new indices for the detection of deception. *Journal of Psychophysiology* 40.3: pp. 239-245.

Grauman K., M. Betke, J. Lombardi, J. Gips, G. Bradski (2003). Communication via eye blinks and eyebrow raises: video-based human-computer interfaces. *Universal Access in the Information Society* 2.4: pp. 359-373.

Han, Y.J., W. Kim, and J.S. Park (2018). Efficient eye-blinking detection on smartphones: A hybrid approach based on deep learning. *Mobile Information Systems.*

Jongkees, B.J. and L.S. Colzato (2016). Spontaeous eye blink rate as a rpedictor of dopamine-related cognitive function- A review. *Neuroscience Biobehavioral Review* 71: pp. 58-82.

Karson, C.N. Spontanous eye-blink rates and dopaminergic systems (1983). *Brain: A Journal of Neurology* 106.3: 643-653.

Karson, C.N., P.A. Lewitt, D.B. Calne, and R.J. Wyatt (1982). Blink rates in parkinsonism. *Annals of Neurology* 12.6.

Lee, W.O., E.C. Lee, K.R. Park (2010). Blink detection robust to various facial poses. *Journal of Neuroscience Methods* 193.2: pp. 356-372.

Li, Y., M.C. Chang, and S. Lyu (2018). In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking. *IEEE Workshop on Information Forensics and Security (WIFS).*

Marcos-Ramiro, A., D. Pizarro-Perez, M. Marron-Romera, D. Gatica-Perez (2014). Automatic blink detection towards stress discovery. *16th International Conference on Multimodal Interaction:* pp. 307-310.

Pan, G., L. Sun, and Z. Wu (2007). Eyeblink-based anti-spoofing in face recognition from a generic webcamera. *11th IEEE International Conference on Computer Vision*: pp. 1-8.

Sun, Y., S. Zafeiriou, M. Pantic (2013). A hybrid system for online blink detection. *Hawaii International Conference on System Sciences.*

Taylor, J.R., J.D. Elsworth, M.S. Lawrence, J.R. Sladek Jr., R.H. Roth, and D.E. Redmond Jr. (1999). Spontaneous blink rates correlate with dopamine levels in the caudate nucleus of MPTP-treated monkeys. *Experimental Neurology* 158.1: pp. 214-220.

Trutoiu, L.C., E. J. Carter, I. Matthews, and J. K. Hodgins (2011). Modeling and animating eye blinks. *ACM Transactions on Applied Perception* 8.3: pp. 1–17.

Wilson, G (2002) . An analysis of mental workload in pilots during flight using multiple psychophysiological measures. *Journal of Aviation Psychology* 16.4: pp. 258-265.

Wu, J. and M.M. Trivedi (2010). An eye localization, tracking and blink pattern recognition system. *ACM Transactions on Multimedia Computing, Communications, and Applications* 6.2: pp. 1-23.