AutoNotes: A Touch-Free Blink-Based Interactive Model for Generation of Notes from Lecture Videos

Pragma Kar1 • Soumya Banerjee1 • Sandip Chakraborty2 • Matangini Chattopadhyay3

Abstract The recent explosion of interest in online courses can directly be attributed to the current pandemic which has collated to the adoption of Massive Open Online Courses. While the enforced social distancing protocol inevitably demands the involvement of technology in raising awareness, it further requires the imposition of preventive restrictions on touch-based systems. Visual gestures like simple blinks can account for a significant subset of possible actions in mobile application domains. The proposed model introduces a novel approach that facilitates the automated generation of relevant lecture notes and web-linked keywords from videos through a blink-controlled interface. Functions like auto-video pausing and auto-closing of the application based on eye attribute tracking have been integrated into the system for handling context switching. The experimental results reveal 84.17 usability score, an average of 89.55% of different types of blink detection accuracy, 91.56% of text generation accuracy and 70.41% of keyword detection accuracy along with low false positives and false negatives under different lighting conditions.

Keywords MOOC • Blink-based interaction • Notes generation • Salience detection

Introduction

Massive Online Courses (MOOCs) have gained enormous popularity due to the evolution of traditional learning from centralized classrooms to global knowledge distribution. Moreover, the unforeseen pandemic situation due to the COVID-19 virus has led to the avoidance of mass gathering and limiting physical contact among individuals and devices. The inflation in the utility of online courses is also a direct result of this scenario.

However, in pre-recorded video lectures, the pace of delivery of the contents does not match with that of cognition for every student. Even in a real-time presentation, it is often impractical to discuss each of the personal queries. In the case of confusion, the learner often takes note of the important points so that the queries can be mediated later. The learner might often look into important terms discussed in a lecture and find relevant materials from other online resources. These may include keywords or figures shown in a video. While key terms can be noted quickly, noting down figures or descriptions while the online video is being played, requires repetitive pausing and playing involving physical contact with the device. This is further infeasible if the course is being followed by the learner in a mobile scenario. Since the process of taking notes for relevant article search is essential, the mentioned intricacies necessitate an interface that does not require physical contact like button clicks and can be controlled.
through facial cues for selecting the sections from the lecture and generate relevant links for the keywords.

However, such a model needs to address generic problems like the free head movement, context switching and restricted set of gestures to be used for controlling the interface. In this paper, a novel, automated note-generation system has been proposed which can be controlled through simple blinks. The model is an integration of simultaneous eye tracking from the user’s preview, notes selection and image capture based on the blink counts. This is followed by a speech to text translation, keyword retrieval and generation of Wikipedia links for the keywords. A video instance can also be paused, captured or closed by blinks or gazing down. The contributions of the paper are as follows:

Firstly, the paper depicts a novel interactive yet contact-free video control system that utilizes user’s blink counts for selection of lecture segments and automatic generation of textual notes and diagrams from these sections. The contact-free interface makes it ideal for the current pandemic situation. Secondly, AutoNotes reduces the reference search time for learners by eliminating the requirement of manually marking keywords or searching them in Web. The system identifies keywords and links them to the corresponding Wikipedia links without any manual intervention. Thirdly, AutoNotes is a client-side system that runs on the user’s device and does not store any user-centric data like facial image. User’s privacy is maintained throughout execution of the system. Fourthly, the system is invariant to different lighting conditions and is highly user friendly, requiring no prior practice or expertise in this unconventional video control technique. Lastly, since AutoNotes deals with video lectures, it seamlessly handles inevitable activities like temporary context switching or multitasking, free head movement and involuntary blinks, without affecting is performance. The approach is lightweight and requires no additional hardware other than a webcam, which is available for most of the devices. It is hence suitable for devices like desktops and laptops.

Related Work

Facial gestures have been largely contributing to the research domain involving almost every aspect of life. While facial gestures largely involve facial emotion detection [1] for a wide range of applications starting from medical condition detection [2] to cognition estimation [3], eyes solely can contribute majorly to these purposes. Gaze and gaze gestures [4] are widely used in ubiquitous computing applications but require extensive continuous tracking for significant performance. Eye blinks [5] are much simpler to identify through different approaches like color appearance [6], feature-based [7] and neural network-based models [8] to contribute in fields like transport safety by identifying fatigue level of drivers [9], contact free mobile/computer interactions [10][11] primarily to help the physically challenged individuals, security domain [8] and so on. However, the application of blink-based human computer interaction in the domain of academia is still nascent.

Voice recognition is a prerequisite for tasks like voice search. Google developed a very rich language recognition model [12]. Schalkwyk et. al. presented a study [13] on Google Search by Voice and demonstrated its accuracy. However, voice commands in mobile environments might get distorted due to facial obstructions like masks. Translation of pre-recorded voice to texts can be feasible in such scenarios. Text processing can be executed from images through using optical character recognition [14].

Identifying the salient concepts for a text has been an open challenge for many years. The classical approach had been to identify the significant keywords in the text utilizing approaches like POS tags, n-grams, etc. [15][16][17]. However, the limitation of this approach is that oftentimes the actual salient concept is not mentioned in the text but must be inferred from the contextual information. To address this limitation, it is common to utilize an external knowledge base to infer the contextual information. TAGME [18] is a very widely accepted entity linker that identifies and links text topics to Wikipedia entries. WAT [19] is an improvement to the TAGME pipeline that significantly improves its performance. However, neither TAGME nor WAT is designed for the entity salience task. What this means is that [18, 19] cannot tell that in the context of machine learning “CNN” refers to “Convolutional Neural Network” and not “CNN” the news network. This is a rather severe limitation for use in niche domains. Like in academic computer science contest “CNN” seldom refers to the news network, but in Wikipedia, on which TAGME is trained, that use is overwhelmingly dominant. Salience detection attempts to address this limitation. SWAT extends on the WAT approach to propose a state of the art salience detection [20]. While the performance of SWAT is not perfect, as we show in the experimental results, for our task it is mostly adequate.
Proposed Model

This section discusses the functional components of AutoNotes according to their work and workflow. The software details and sub-modular components are also analyzed in this section.

Functional Modules of AutoNotes

In this section, a detailed discussion of the proposed model has been presented, along with the discussion of its individual modules. Figure 1 depicts the overview of different architectural components of the system. The following subsections present the details of the blink-based control module and the section processing and notes generation module, as shown in Fig. 1.

Blink-based control module

The execution of the system begins with this initial module which integrates a sequence of hierarchical condition checking mediated through continuous eye tracking and corresponding lecture video execution. The control actions that can be performed by the user to control the selection and viewing process are as follows:

Double blink: Each double blink marks the beginning and end of a note section in an alternate pattern. If the user forgets to end the last section selection, the end point is...
taken as the end point of the video. For each section, a sectional note will be generated after the program closes.

**Triple blink:** In some cases, the user might want to freeze a particular frame, e.g., a diagram, to follow further references to the diagram in the later parts of the video. In the proposed model, a learner can perform a triple blink when a diagram is shown in the video (or at any frame). This will capture the frame and display it in a window for reference, while the video lecture plays in the window beside.

**Gaze down/ Moving out of the frame:** If the learner temporarily looks down, closes his/her eyes for a few seconds or becomes temporarily unavailable, the video lecture will automatically pause until the learner looks back at the screen or moves back in front of the screen. This is to ensure that periodic context switching is supported by the system. However, each time the user looks down/ stays out for 5 more seconds, the program closing counter starts. At 5th second, the user is prompted with a voice command. If the user continues to look down/ stay out for 5 more seconds, the program closes. This ensures a contact-free exit mechanism at any instance of the lecture. Assuming an average frame rate of 15fps, the program will be closed if eyes are closed for 150 consecutive frames. The threshold of eye aspect ratio (EAR) to infer if the eye is closed, is set to .15. Mathematically, if \(\text{EAR}(\text{Frame}_{n}(\text{pre}), ..., \text{Frame}_{n+150}(\text{pre})) < .15(\text{Thresh})\), the program is closed. However, if \(\text{EAR}(\text{Frame}_{n}(\text{pre}), ..., \text{Frame}_{n+k}(\text{pre})) < \text{Thresh}\) such that \(k < 150\), then display \(\text{Frame}_{n}(\text{lec})\) \(\forall \text{Frame}(\text{pre}) \in \{\text{Frame}_{n}(\text{pre}), ..., \text{Frame}_{n+k}(\text{pre})\}\), afterwards \(\text{Frame}_{n+1}(\text{lec})\) is displayed. Here, \(\text{pre}\) and \(\text{lec}\) refer to the preview video displaying the user’s face and the lecture video, respectively.

If the preview contains the face of the user, the eye region is detected using the eye landmarks and the EAR is calculated [21]. Let \(C1, C2\) be the landmarks present at the two eye corners. \(UL1, UL2\) be the landmark points on the upper lid and \(LL1, LL2\) be the landmarks on lower lid. Then, \(\text{EAR} = (\text{dist}(UL1, LL1) + \text{dist}(UL2, LL2))/(2 \times \text{dist}(C1, C2))\) where \(\text{dist}(\cdot)\) is the Euclidean distance. The double and triple blink functionality of this module has been depicted in Algorithm 1. The threshold for counting a valid consecutive blink is set as 20 frames for restricting false positives or false negatives. Logically, a triple blink will always be preceded by a double blink. Since double blink invokes the start/stop function in an alternate pattern, a triple blink will always start or stop the selection process before capturing the frame. This is not particularly a problem if the triple blink invokes a section selection start. However, if the selection has already been started, a triple blink will prematurely end it. To avoid this error, a temporary stop function has been added to the system. When a double blink is encountered for an active section selection, the next 20 frames are checked for a valid blink while keeping the stop function temporary. If a blink is found within the window of the next consecutive 20 frames, the temporary closure is undone and the current frame is captured/frozen. However, finding no blinking within the consecutive window stops the session permanently. At the end of every committed stop function, the section is stored for further processing. Varying these thresholds and making them adaptive to the users is a future scope of the work.
Section processing and notes generation module This module is a sequential flow of operations performed on the total set of sections. For each section, the frame numbers are mapped to the corresponding time of the video. From each extracted frame in a section, the written words (if any) are extracted using optical character recognition. This is based on the assumption that in lectures, keywords are often written on boards by the lecturers while explaining the details verbally. For the audio sections, the audio-to-speech module generates the corresponding set of texts (notes).

The final sub-module identifies the keywords from each text section and provides the related Wikipedia links to the learner. The proposed system leverages the advantage of the salient concept annotator, SWAT [20], successor to the widely popular annotator, tagme [18], to identify the conceptual entities from their mentions in the generated notes. For example, in the sentences “The current pandemic has increased the demand for MOOCs...” the module correctly identifies “pandemic,” “the demand” and “MOOCs” as the mention of the conceptual entities “Pandemic,” “Video on demand” and “Massive Open Online Course”, respectively. The module then hyperlinks the mentions to the Wikipedia article corresponding to the conceptual entity. Since Wikipedia articles generally are written to be comprehensible with minimal prerequisite knowledge but have adequate references for further detailed perusal, we feel that such a convenient link in the generated notes will greatly magnify its usefulness to the user.

Algorithm 1: Blink based control

Algorithm 1: Blink based control

J. Inst. Eng. India Ser. B (December 2021) 102(6):1157–1166 1161
Software Architecture and Sub-modular Components of AutoNotes

In this section, each of the component is divided into finer sub components to describe their interactions. Figure 2 depicts the software architecture of the components through the major function calls used in each functional sub-module. In this figure, the input to each module is the output of the previous module. The details of each of these components and their main/important function calls are described next.

**Landmark detection.** Prior to this module the RGB frames containing the facial region of the user are converted to grayscale image by using the OpenCv’s cvtColor(). To identify the facial region of the image from the entire frame, dlib’s get_frontal_face_detector() function is used. This function facilitates face detection by using histogram of oriented gradients (HOG) and linear support vector machine [22]. The shape_predictor() of dlib then allows the identification of 68 facial landmarks from the detected facial region of the user. This is performed by a pretrained model based on [23] and the iBug 300-W dataset¹ for training.

**Blink detection & counting.** This sub-module receives the 68 landmarks from the previous one and prunes them down to select only the landmarks on the top lip, bottom lid and corners. These selected coordinates are passed to the custom EAR_checker() function that identifies whether a blink is performed by estimating the EAR and comparing it with the threshold. Depending on whether this function returns a true or false, the blink_counter() functions modify the blink count. Last registered blink also affects this variable access. In this function, a frame counter variable is also modified if the consecutive frames register absence of eyes or a low EAR to decide whether to pause or exit the video as discussed in the next subsection.

**Video control.** As discussed earlier, video control action depends on the number of blinks or gaze down. Depending on the number of blinks, the start(), stop() or capture() function is called to start the video section selection, stop the selection or freeze a frame, respectively. Gaze down time, depicted by the frame counter variable affects the pause(), resume() or exit() function to halt, proceed with or stop the video, respectively.

**Notes generation.** This module is an integration of OCR-based text and voice processing functions. On exit(), the frames of the recorded segments are processed by the pytesseract function image_to_string() to identify textual

¹ [https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/](https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/)
content from the image. The audio segments are simultaneously processed by the creation of a SpeechRecognition
object using Recognizer() function, loading the audio segment (AudioFile()), and converting it to text using the
Google Speech API (recognize_google()).

**Keyword extraction & Link generation.** The text obtained from the previous module is encapsulated and
sent as a post-request to the rest API for SWAT [20] inside query_SWAT(). SWAT processes a paragraph of text and
identifies the salient concepts mentioned in the text. The response contains the mentions of the word that corre-
sponds to the keywords, its positions and the unique wiki_id corresponding to the salient keyword. The anno-
tate() method iterates over the text and replaces each mention with a hyperlink to its corresponding Wikipedia
page. Further mathematical details are excluded due to space constraint.

**Experimental Results**

**Implementation Details and Devices Used**
The system has been tested on a two desktops and a laptop. The devices had Intel Core i5-4440 CPU @ 3.10GHz \times 4
processor (Ubuntu), Intel Core i5-4670 CPU @ 3.4GHz \times 4 processor (Windows) and Intel Core i5-7200U CPU @ 2.50GHz\times 4 processor (dual). The python libraries like dlib, SpeechRecognition and pytesseract are used for the
development of the system. To further analyze the scalability of the approach, the blink detection, counting and
video control modules were implemented in Android Studio. The application worked in real-time when tested on
devices with Android 10, 6GB RAM, Qualcomm SDM660 octa core processor and Android 8.1.0, Samsung Exynos 7
Octa 7870 processor and 1GB RAM. Implementation of the rest of the non-real-time modules to a smartphone
platform is hence feasible and is a work in progress.

**Subjective Evaluation for Modular Analysis**
This section describes the study designs and results of the experiments according to the modular sub-components of
AutoNotes.

**Method I : System Usability Study**
The first study aimed at estimating the usability of the system in the real world. The standard system usability
scale (SUS) [24] was used for this purpose on a total of 21 participants, 11 females and 10 males, belonging to the age
group of 25–70 years after they watched the demonstration of the system.

**Method II : Subjective evaluation**
The performance of individual modules was tested by performing a set of evaluations on real world users, similar to [25]. It is to be noted that the landmark detection module uses a well established technique with significant perfor-
mance. For the other modules, the results are obtained by the three studies, conducted with 5 participants, 2 females
and 3 males belonging to the age group of 25–60 years.

**Study 1:** In the first study of this session, each particip-
ant was asked to use the system under two different light
conditions: low and normal. A set of predefined blink
patterns (a double blink followed by a triple blink followed
by a voluntary (V) single blink) were given to the users
which had to be performed by them at short intervals to
time with no additional constraint. The given pattern had to
be repeated for 10 times under each lighting condition.
Hence, each participant had to blink for 120 times along
with any additional involuntary blink (InV).

**Study 2:** In this experiment, the participants were asked
to write some words on boards or papers using either a pen,
sketch pen or board marker. Different colored inks were
chosen and the experiment was conducted under normal and
low light conditions. The sentences written were either in
upper case or sentence case. These board and paper
writings were recorded from different distances to create a
collection of 7 videos showing recorded board works. Since
some lecture videos display texts written on papers, the
paper writings are also included.

**Study 3:** In this study, the keyword detection and
overall system performance has been estimated. In this
study, 10 sections were selected from 5 different lecture
videos from the computer science (CS) domain. Each
section was approximately a few minutes long. The tran-
script corresponding these sections were obtained and then
manually annotated by two post-graduates from the CS
domain, independently. The annotations typically consisted
of marking the keywords from the given texts. These were
then validated by two doctoral researchers from the same
domain. Five users were asked to select these predefined
sections from the video lecture using double blinks. The
generated textual note and the keywords were validated
with the previously created ground truth.

**Result 1 : System Usability Study**
The survey revealed an average score of 84.17 for the
system, thus proving its feasibility. Figure 3a shows the
distribution of the scores. It is to be noted that for each
question, the participant can provide a score between 1 and
5. Question-wise scores (Q1-Q10) for each participant are
considered by a formula \(((Q1 - 1) + (5 - Q2) + (Q3 -
1) + (5 - Q4) + (Q5 - 1) + (5 - Q6) + (Q7 - 1) + (5 -
Q8) + (Q9 - 1) + (5 - Q10)) \times 2.5\) to get a scaled SUS
score. The average question wise scores are shown in
Fig. 3b. The questions are provided in such an order that

\footnote{swat.d4science.org}
getting higher score for odd numbered (positive statements) questions and lower scores for even numbered ones (negative statements) result in higher scaled score.

**Result II : Modular Analysis through subjective evaluation**

The results of Method II are discussed in this section, with respect to the components of AutoNotes.

**Study 1 : Result for Blink detection, counting & Video control**

Figure 4a, b shows the comparison between the average misses (false negatives) and number of extra blinks detected (false positive) under the two lighting conditions. The results show that the impact of light is negligible on the performance.

The missed count can be interpreted as the V and InV that go undetected. For double and triple blinks missed indicate the number of double or triple blinks that were not/mis-registered. False positive accounts for the total false blinks that are generated by the system (total blinks). For double, triple blinks, it is the additional number of false triggers that results in additional section creation or image capture. It is noted that most of the false positives are single blinks, thus resulting in no additional section or capture. However, in case of some participants, sets of falsely generated blinks had falsely triggered triple blinks and hence additional image capture. Analyzing the total intended blinks and correctly predicted blinks in each category, the accurate blink prediction rates are shown in Fig. 4c. Total blink accounts for the correctly detected Single(V) + Double * 2 + Triple * 3 + InV. The total accuracy of our system (97.5%), when compared to standard blink detection accuracy [26], shows improved results.

Even though involuntary blinks are registered by the system, they had no practical impact on the function calls.
This is due to the fact that in the implementation of the approach, some criteria were set. Firstly, if the second double/ triple blink occurs immediately after the first, the session start is reversed/ capture is not registered. Secondly, exactly two/ three blinks will invoke a function call. If a sequence of four (say) blinks are detected due to false positives or user’s involuntary blink, the fourth blink will not have an impact.

**Study 2 : Result for Notes generation**

The videos recorded in Study 2 were analyzed separately by the OCR function, `image_to_string()` for the detection of handwritten texts and their conversions. Table 1 presents the percentage of correctly detected characters by grouping the videos into categories like light, case of the written texts, recorder distance, pen type and color. The results are estimated based on the different parameters to test the scalability of the module. The percentages vary from acceptable to excellent by showing the highest accuracy under normal light and lowest for writing with pen. The detection accuracy for higher distance is low due the reflection of light on board area for 2 videos.

For the speech to text generation function, results showed that 91.56% of the transcripts were generated correctly.

**Study 3 : Result for Keyword extraction & Link generation**

70.41% of the total keywords from all the videos were identified and linked to Wikipedia in proper contextual form. The individual video-wise accuracy of keyword detection and link generation for the 5 videos were found to be 72.7%, 100%, 57.14%, 70.58% and 81.25%, respectively.

**Conclusion**

The paper presents a novel approach that facilitates touch-free interaction with a video lecture to automatically generate relevant study materials and reference links based on the user-selected sections. The use of simple blink gestures facilitates easy handling of the system features by the learners. Even though experimental results demonstrate the usefulness of the proposed system, future directions will aim at improving the note generation accuracy. Moreover, board occlusion removal is a feature that can promote the quality of the lecture video. Even though freezing frames for reference is a solution for occluded boards in frames, by extracting foreground pixels and extrapolating background pixels, occluded boards can be recovered. The usability scale proves the feasibility and necessity of such a system in the practical domain and hence establishes its promising scope.

**Acknowledgements** This work has been supported by Innovation in Science Pursuit for Inspired Research (INSPIRE) program of the Department of Science and Technology (DST) (PhD Program). We thank all the participants for their consent and participation in the user studies.

**Funding** The research has been supported by and conducted under the funding provided by the Innovation in Science Pursuit for Inspired Research (INSPIRE) program of the Department of Science & Technology (DST) (Ph.D. Program).

**Compliance with Ethical Standards**

**Conflicts of interest** The authors declare no conflict of interest.

**References**

1. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-workshops, pp. 94–101, IEEE (2010)

2. L.A. Rutter, D. Dodell-Feder, I.V. Vahia, B.P. Forester, K.J. Ressler, J.B. Wilmer, L. Germine, Emotion sensitivity across the lifespan: mapping clinical risk periods to sensitivity to facial emotion intensity. J. Exp. Psychol. Gen. 148(11), 1993–2005 (2019)

3. M. Mukhopadhyay, S. Pal, A. Nayyar, P.K.D. Pramanik, N. Dasgupta, P. Choudhury, Facial emotion detection to assess learner’s state of mind in an online learning system, in Proceedings of the 2020 5th International Conference on Intelligent Information Technology, pp. 107–115 (2020)

4. P. Kar, S. Chattopadhyay, S. Chakraborty, Gestatten: estimation of user’s attention in mobile moocs from eye gaze and gaze gesture tracking, in Proceedings of the ACM on Human–Computer Interaction, vol. 4, no. EICS, pp. 1–32 (2020)

5. A. Królak, P. Strumillo, Eye-blink detection system for human–computer interaction. Univers. Access Inf. Soc. 11(4), 409–419 (2012)

6. A. Panning, A. Al-Hamadi, B. Michaelis, A color based approach for eye blink detection in image sequences, in 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 40–45 IEEE (2011)
7. T. Morris, P. Blenkhorn, F. Zaidi, Blink detection for real-time eye tracking. J. Netw. Comput. Appl. 25(2), 129–143 (2002)
8. Y. Li, M.-C. Chang, S. Lyu, In ictu oculi: Exposing ai created fake videos by detecting eye blinking, in 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7, IEEE (2018)
9. T. Danisman, I. M. Bilasco, C. Djeraba, N. Ihaddadene, Drowsy driver detection system using eye blink patterns, in 2010 International Conference on Machine and Web Intelligence, pp. 230–233, IEEE (2010)
10. T. Ohno, N. Mukawa, S. Kawato, Just blink your eyes: a head-free gaze tracking system, in CHI’03 Extended Abstracts on Human factors in Computing Systems, pp. 950–957 (2003)
11. A.P.A.A. Mohammed, S.A. Anwer, Efficient eye blink detection method for disabled-helping domain. Eye 10(P1), P2 (2014)
12. O. Hurst-Hiller, J. Farago, Searching for content using voice search queries, Mar 2 2010. US Patent 7672931
13. J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, B. Strope, your word is my command: Google search by voice: A case study, in Advances in Speech Recognition, pp. 61–90, Springer (2010)
14. S.R. Chowdhury, P. Kar, M. Chattopadhyay, M. Bhattacharya, S. Chattopadhyay, Mobile enabled content adaptation system for pdf documents, in 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1–8, IEEE (2017)
15. Z. Liu, W. Huang, Y. Zheng, M. Sun, Automatic keyphrase extraction via topic decomposition,” in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 366–376 (2010)
16. P.D. Turney, Learning algorithms for keyphrase extraction. Inf. Retr. 2(4), 303–336 (2000)
17. D. Paranjpe, Learning document aboutness from implicit user feedback and document structure, in Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 365–374 (2009)
18. P. Ferragina, U. Scaiella, Tagme: on-the-fly annotation of short text fragments (by wikipedia entities), in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1625–1628 (2010)
19. F. Piccinno, P. Ferragina, From tagme to wat: a new entity annotator, in Proceedings of the First International Workshop on Entity Recognition and Disambiguation, pp. 55–62 (2014)
20. M. Ponza, P. Ferragina, F. Piccinno, Swat: a system for detecting salient Wikipedia entities in texts. Comput. Intell. 35(4), 858–890 (2019)
21. J. Cech, T. Soukupova, Real-time eye blink detection using facial landmarks, Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague, pp. 1–8 (2016)
22. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, pp. 886–893, IEEE (2005)
23. V. Kazemi, J. Sullivan, One millisecond face alignment with an ensemble of regression trees, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874 (2014)
24. J. Brooke, A Quick and Dirty Usability Scale. Usability Evaluation in Industry (Taylor & Francis Ltd, London, 1996).
25. M. Chau, M. Betke, Real time eye tracking and blink detection with USB cameras, tech. rep., Boston University Computer Science Department (2005)
26. M. Wang, L. Guo, W.-Y. Chen, Blink detection using adaBoost and contour circle for fatigue recognition. Comput. Electr. Eng. 58, 502–512 (2017)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.