A Toolkit for the Automatic Analysis of Human Behavior in HCI Applications in the Wild

Article · January 2020

DOI: 10.25046/aj050622

5 authors, including:

Andrea Generosi
Università Politecnica delle Marche
10 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Silvia Ceccacci
Università Politecnica delle Marche
53 PUBLICATIONS **151** CITATIONS
SEE PROFILE

Luca Giraldi
Università Politecnica delle Marche
15 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Maura Mengoni
Università Politecnica delle Marche
134 PUBLICATIONS **536** CITATIONS
SEE PROFILE

Advances in Science, *Technology and Engineering Systems Journal, Vol. 5, No. 6, 185-192 (2020)*

www.astesj.com

Special Issue on Multidisciplinary Innovation in Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

A Toolkit for the Automatic Analysis of Human Behavior in HCI Applications in the Wild

Andrea Generosi[1,*], Silvia Ceccacci[1], Samuele Faggiano[2], Luca Giraldi[2], Maura Mengoni[1]

[1]*Department of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, Ancona, 60131, Italy*

[2]*Emoj s.r.l., Ancona, 60131, Italy*

ARTICLE INFO

Article history:
Received: 03 August, 2020
Accepted: 07 September, 2020
Online: 08 November, 2020

Keywords:
User Experience
Deep Learning
Convolutional Neural Networks
Affective Computing
Gaze detection

ABSTRACT

Nowadays, smartphones and laptops equipped with cameras have become an integral part of our daily lives. The pervasive use of cameras enables the collection of an enormous amount of data, which can be easily extracted through video images processing. This opens up the possibility of using technologies that until now had been restricted to laboratories, such as eye-tracking and emotion analysis systems, to analyze users' behavior in the wild, during the interaction with websites. In this context, this paper introduces a toolkit that takes advantage of deep learning algorithms to monitor user's behavior and emotions, through the acquisition of facial expression and eye gaze from the video captured by the webcam of the device used to navigate the web, in compliance with the EU General data protection regulation (GDPR). Collected data are potentially useful to support user experience assessment of web-based applications in the wild and to improve the effectiveness of e-commerce recommendation systems.

1. Introduction

This paper is an extension of work originally presented in ISCT [1]. Understanding User eXperience (UX) in the wild is of paramount importance for online business (e.g., e-commerce) to gain information about users' opinion and behavior, in order to increase productivity, customize services and better drive strategic decisions. According to the definition provided by the international standard on ergonomics of human-system interaction, ISO 9241-210, UX is "a person's perceptions and responses resulting from the use or intended use of a product, system or service". It follows that the analysis of the sensations and emotions of the user with the product represents a central topic for UX research [2]. Over the last decade, increasingly accessible social media and tools offered by e-commerce sites themselves (e.g. Amazon.com), which encourages users to share their opinions, have made available a wealth of information potentially useful for understanding UX. Alongside, there has been a growing interest in sentiment analysis based on text analysis and user opinion mining. Several automatic approaches have been proposed to understand user emotion and extract UX information from online user online review [3-5]. However, though these tools are becoming increasingly accurate and effective, it is impossible to rely only on online reviews to understand how customers really feel. As reported by Jakob Nielsen in a study about user behavior in online communities [6], a significant number of customers (90%) are lurkers (i.e., they read and observe, but they do not contribute), while the 9% contribute intermittently within the only community and only the 1% account for most contributions. This phenomenon has been named "the 90-9-1 rule". Consequently, while systems able to get direct feedback from all the users (whether they are someone who will leave on online review or not) are still lacking, companies continue to spend millions of dollars on surveys and analysis of their customers to know what really works and what does not.

Moreover, recent years have shown an increasing interest in enhancing possibilities of human-computer interaction with e-commerce. Recommendation systems have started to attract interest in both business and research [7]. In the last few years there was significant improvement especially over the collaborative filtering approaches thanks to the advance in the field of machine learning and deep learning techniques [8]. However, the effectiveness of such a system remains generally low, as the algorithms used by most of the web e-commerce for providing shopping suggestions to customers use data collected from other customers, so that system difficulty meets the

*Corresponding Author: Andrea Generosi, a.generosi@pm.univpm.it

https://dx.doi.org/10.25046/aj050622

customer's expectation [7]. In this context, emotion detection represents one of the directions that can be taken to enhance the overall UX with these systems. In fact, it has been proved that recommender systems are more precise in providing relevant suggestions, when emotion variables are considered [9].

Nowadays, a variety of technologies exists to automatically recognize human emotions, spanning across facial expression analysis, acoustic speech processing and biological responses interpretation. The extensive deployment of smart devices equipped with sensors and connected to the Internet today opens the possibility of accessing a huge amount of data that until now would have been very difficult to collect. Smartphones and laptops equipped with cameras became an integral part of our daily lives. The use of cameras is so pervasive that we no longer worry about their presence, and we are not fully aware of how much information can be collected through them. For example, using appropriate systems, it is possible to process video images so that demographic (e.g., gender and age) and behavioral (e.g., emotions, eye movements) information can be easily extracted [10, 11]. This opens up the possibility of using technologies that until now had been restricted to laboratories to analyze UX in the wild.

In this context, this paper aims to extend and integrate the contribution of the work presented in ISCT, which introduced and evaluated a new system of emotion detection for mobile applications, in order to define a toolkit for automated collection of data related to users' emotions and behavior useful to:

- Support the collection of data related to the user behavior during the interaction with desktop and laptop web-based applications in the wild;
- Improve the effectiveness of recommendation systems.

Such a tool will take advantage of the system proposed in [1] to monitor user's emotions, through the acquisition of facial expression and implements new system based on deep learning algorithms to track the eye gazes from the video captured by the webcam of the device used to navigate the web, in compliance with the General data protection regulation (GDPR), the European Union regulation on the processing of personal data and privacy.

The remainder of this paper is organized as follows: Section 2 gives an overview of the state of art in the context of systems for the automatic detection of users' emotions and behavior and in particular of facial expression recognition and gaze tracking systems. Section 3 describes the overall system architecture with the Emotion recognition and Gaze detection modules. In Section 4 performance evaluation results of the proposed are reported. The last section summarizes the main paper contributions and highlights future outlook.

2. Research Background

Over the past years producers and the marketing/branding industry demonstrated an increasing interest in emotional aspects of user behavior, as understanding the emotional state of users is crucial for developing successful products and services [12]. Psychophysiological methods may offer data throughout the process of experience, which unfolds new possibilities for UX evaluation. This motivated the demand for automatic emotion recognition techniques as a tool for getting a larger quantity of more objective data. Several attempts have been made to explore the possibility to use sentiments analysis to extract UX information from online user reviews. However, online reviews have proved to be too generic and lacking contextual references to effectively support UX assessment [13]. Several studies, such as [14] and [15], proposed systems that allow us to correlate data from different typologies of data, e.g. eye-tracking fixations, sentiment analysis, body gestures, or facial expressions. However, these systems are designed to support only formal UX evaluation assessment, while are not suitable for studies in the wild. Based on the best of our knowledge, no studies so far have proposed the integrated use of gaze and emotion recognition systems, based on deep learning algorithms, to support the collection of relevant information useful for UX assessment of laptop and desktop web applications nor to power recommending systems. Only [11] proposed a system to support UX assessment based deep learning, but such a solution works only for mobile applications. While only [7] proposed to process data provided by a common pc webcam to enable users' gaze and emotion detection in order to manage an intelligent e-commerce recommendation system: it proposed respectively to use SVM algorithm to enable emotion recognition and a gradient-based method to perform gaze tracking. These algorithms, compared with most of Deep Learning CNNs, are more efficient for what concerns the computational performance, but less accurate [16].

2.1. Emotion recognition

In recent decades, research in the field of Human-Computer Interaction has shown an increasing interest in topics such as Affective computing, emotion analysis and study of human behaviors.

This has encouraged the development of numerous methods and tools for the recognition of human emotions, characterized by different levels of intrusiveness. Most of them relate to three main research areas: facial expression analysis, speech recognition analysis and biofeedback analysis. Instruments based on biofeedback currently available (e.g., ECG or EEG, biometric sensors) are still difficult to adopt for studies in the wild, although the use of smart watches and bracelets is increasingly widespread [17]. Instead, research efforts in the field of facial expression recognition systems have allowed the development of reliable and non-invasive systems. The theoretical model most widely used in the analysis of facial expressions to develop algorithms for the recognition of emotions is certainly represented by the Face Action Coding System (FACS) [18]. It allows the identification of six universal emotions (i.e., joy, surprise, sadness, anger, fear and disgust) by tracking the movements of the face muscles.

The vast majority of facial expression recognition systems are based on Convolutional Neural Networks (CNN): a particular mathematical model of Deep Learning that, unlike classical neural networks, present some layers, in the first part of the entire network, which apply a convolution instead of a general matrix multiplication to the images in input before passing the result to the next layers [10]. Several tools have been proposed in literature, such as [19-21]. Usually these systems refer to trained models using datasets built in controlled environments, where it

is possible to obtain the best accuracy scores, or, trained with data obtained by crawlers on the web, with low accuracy but mostly reflecting real contexts. To the best of our knowledge, approaches able to ensure a good accuracy using data obtained "in the wild" have not yet been evaluated. To ensure good accuracy in the recognition of human emotions in different contexts of use, the system of recognition of emotions proposed in this study adopts a hybrid approach. The resulting CNN, based on Keras and Tensorflow frameworks, has been trained merging three different public datasets.

2.2. Eye-gaze tracking

Tracking one's eyes movement is challenging yet important in HCI and computer vision fields. When it comes to products (as well as website) design, it is readily understandable how much it means to know where a user is looking at. In recent years research has aimed to achieve increasingly accurate results, using less and less invasive and off-the-shelf systems (especially webcams).

According to the results of a recent survey, gaze tracking techniques can be classified into two main categories: feature-based and appearance-based [22]. The first approaches require the acquisition of high-quality images to determine ocular characteristics, such as infrared corneal reflections [23], pupil center [24] and iris contour [25]. To determine the direction of the gaze independently to the head pose, these features are related to 3D eye models. Although such approaches allow high accuracy to be achieved, the accuracy of the geometric calculation is highly dependent on system calibrations that are often difficult or impossible for ordinary users to perform. In addition, as they need to extract small features of the eye, they require the use of special equipment (e.g. IR light sources, special glasses / contact lenses) to acquire special high-resolution infrared images. Such technologies are not normally available in uncontrolled environments. Conversely, appearance-based approaches do not need particular illumination conditions, so that they work well with natural lighting, and require only a single webcam [26]: they take image contents directly in input and try to implicitly extract some relevant features, thus establishing a mapping to screen coordinates. Applicability of this latter kind of methods is remarkably large, although the potential handling of low-resolution images makes their accuracy generally lower [26].

Recently, deep learning (DL) and convolutional neural networks (CNN) have been gaining interest for gaze estimation, so much so that many datasets [27] and network architectures [26, 28, 29] have been proposed over the last years. The gaze-tracking system described in this paper adopts an architecture similar to the one proposed by Krafka et al. [30], which represents one of the more solid CNN actually proposed for gaze tracking. It is based on the AlexNet model described in [31].

3. System Architecture

For the proposed toolkit a centralized architecture has been designed. This kind of architecture has been inherited from the system described in [11] and adapted to be used in synergy with web platforms instead of mobile applications. In Figure 1 are shown the main components: the Web Plugin and the Deep Learning Platform (DLP), respectively the frontend and the backend sides. The Web Plugin is a Javascript library with different functions usable by the developers (APIs) to get info about user's interactions with the web application, i.e. interactions timestamps, clicks and scroll coordinates and above all the webcam handling, so how and when to enable the webcam activation, shooting frequency etc. This feature is the most important because both the Deep Learning engines that compose the DLP make use of the user's face photos to return the information related to his behavior. Photos taken through the pc webcam are encoded in base64 from the Web Plugin and sent through a HTTPS POST to the REST interface exposed through a specific endpoint from the DLP and developed in Python. The received call is then decoded and parsed to get the original data and store it in the server volatile memory, to ensure the compliance of the system with GDPR about privacy. After that, the encoded photo is stored in three different Redis queues: in this way every DL Tracker module can process every photo asynchronously. These queues are used to store and manage the order of arrival of the data that will be given in input to the Gaze Tracker and Emo Tracker for the prediction, respectively, of the gaze coordinates on the screen and the universal Ekman's emotions. Both trackers are Convolutional Neural Networks (CNN) implemented with the Python frameworks Tensorflow and Keras. Whenever the photo processing is concluded for a CNN, the output is saved in a database.

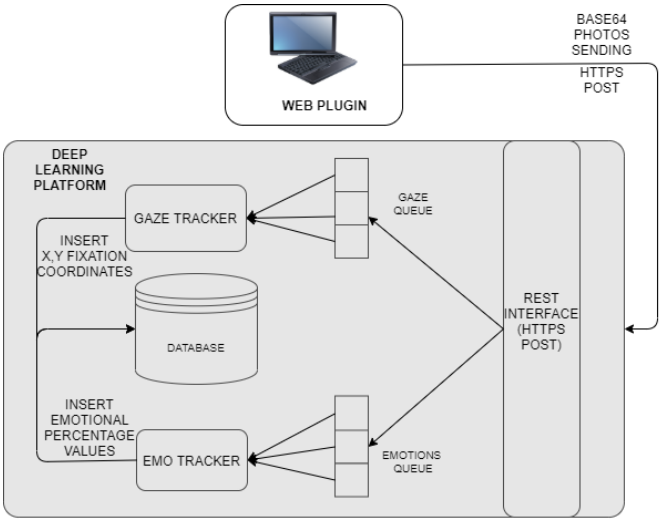

Figure 1: System architecture

3.1. Emo Tracker - the Emotion detection module

Based on [20], a state-of-the-art Deep Learning model that performs well in image recognition should also perform well in a facial expression recognition task: that's because basically the visual discrimination of human emotions is an image classification task. According to different review works, like [19], there exist different DCNNs (Deep Convolutional Neural Networks) that reach various accuracy levels, and those that reach the highest performances are always trained using datasets composed by images retrieved in controlled environments and labeled in labs. Instead, lower accuracy percentages are reached by those datasets with "in the wild" properties, especially retrieved by web crawlers, usually composed by blurred, decentralized or very low-resolution images with misclassified

labels [20]. In return, this kind of datasets are often significantly bigger than the other kind.

To achieve the best emotion recognition accuracy and performances, various tests were conducted. Firstly, by merging the public datasets CK+ (developed in laboratory) [32], FER+ (the FER dataset re-tagged version with crowdsourcing) and AffectNet (manually annotated), a big dataset is constructed. More in detail, FER+ dataset has a label accuracy about 90%, with just 35k photos [20], AffectNet instead has more than 1 million web crawled images, but it also provides more than 400k photos manually labeled by several experts. These datasets have been chosen because they are tagged with almost one of the universal Ekman's emotions labels and because they reach a labeling accuracy relatively high. Moreover, as mentioned, they belong to different dataset categories, so they provide the possibility to improve the resulting model reliability: the assumption is in fact that combining lab generated datasets with those with "in the wild" properties, it is possible to obtain a better accuracy for the in the wild benchmarks. To further improve resulting dataset quality, a Python script to filter all the images with none or more than one face has been developed. The resulting merged dataset at the end counts more than 250k photos, with data distribution shown in Figure 2.

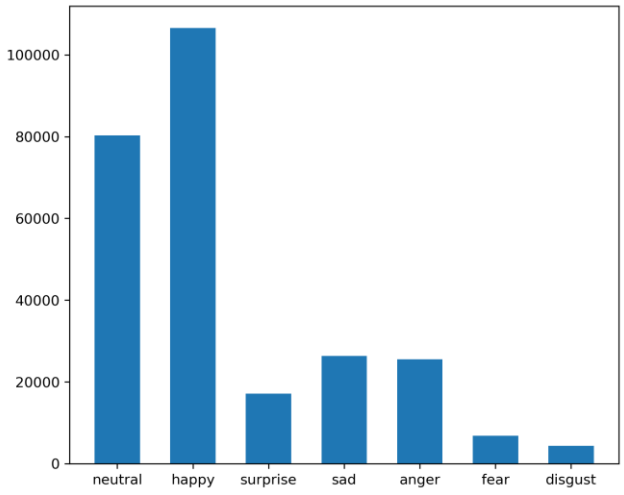

Figure 2: Data distribution

Every single dataset has then splitted into two different parts for training and validation, so to avoid distorting the results during validation phase: for this purpose it was chosen, empirically, to divide each dataset using the ratio 80% for training and 20% for validation, equally for each dataset. After constructing the dataset, preprocessing steps are performed such as facial alignment, centralization of the face respect to the image and face rotation, obtained using Dlib facial landmarks coordinates as reference. All the photos have then been scaled to 64x64 pixels, so to make them homogeneous in size.

For the implementation of the Emo Tracker module, different Python scripts that make use of Dlib, Tensorflow and Keras frameworks have been developed. In particular, Dlib is used to detect one or more human faces in a frame (Face Detection) and provide main face landmarks coordinates, while Keras and Tensorflow provide some of the most used APIs in the Deep Learning field. Trained networks models have been defined to take 64x64 pixels face images in grayscale by the input layer, and to return the Ekman's Emotions (joy, surprise, anger, disgust, sadness, fear and neutral) classification probability by the output layer. Different Keras model architectures such as Inception [33], VGG13, VGG16 and VGG19 [34], were also tested to compare the reached accuracy levels, as listed in Table 1.

Table 1: Performance of different models

Architectures	Accuracy (%)
VGG13	**75.48**
VGG16	74.48
VGG19	73.14
InceptionV2	75.26
InceptionV3	67.20

Network hyperparameters are initialized as it is stated on [20], then the other variations are also experimented such as validation split 0.1, 0.2, number of epochs 30, 50 and 100 and dynamic learning rate defined as

$$lr = lr \times (1 - \frac{epoch}{epoch})$$

Learning rate (lr) is initialized with 0.025 and updated on each epoch accordingly. As shown in table 1, VGG13 model reaches the best accuracy percentage in validation phase, accuracy values over time in relation to the epochs increasing are shown on Figure 3, as a reference of the carried-out experiment. Overfitting occurs during training using other deep neural network architectures, except for the VGG13, when epochs number is bigger than 30.

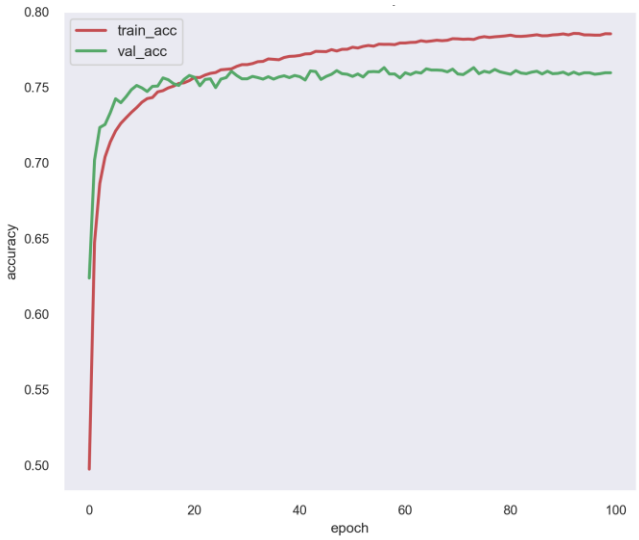

Figure 3: Training and validation accuracy of VGG13

3.2. Gaze Tracker - the Eye-Gaze Detection module

In 2016, Krafka et al. [30] published a paper with a solid CNN proposal for gaze tracking, together with a big dataset of crowdsourcing-collected faces; many researchers are referring to it since then. The Gaze Tracker module's network adopts an architecture similar to the one proposed in [30], which is based on the AlexNet model [31], but respect to that one, a dropout layer

was added to the right after each last convolutional layer. Also in this case, the output is the camera distance, in centimeters. The main challenge using this architecture has been to adapt it to an application context different from the one originally established: iTracker in fact has been trained to work with mobile devices, especially it can reach the best performances on iPhone devices, where it is possible to exactly know, dynamically, the phone camera position. Moreover, practically the whole dataset used to train the model, was built using photos taken by Apple devices. Our purpose, instead, is to propose a gaze detection module optimized to be used on laptop and desktop devices, while it has been decided to not consider a general model that involves also mobile displays in order to not undermine the performance of the model for the specific case taken into consideration by this research.

To achieve this objective, a new training dataset has been built from scratch, collecting pictures from informed volunteers who decided to contribute. For this purpose, a web application has been developed. Each participant was asked to stare at a red dot that was randomly displayed in 30 different screen positions, while the webcam took a picture for each of them; corresponding point coordinates were stored too, to ensure image-screen coordinates association. Concerning screen sizes, this application has been thought to be as general as possible (i.e., capable of dealing with different kinds of displays). However, due to infrastructural reasons, it is not possible to automatically retrieve physical dimensions of a screen from a web page application.

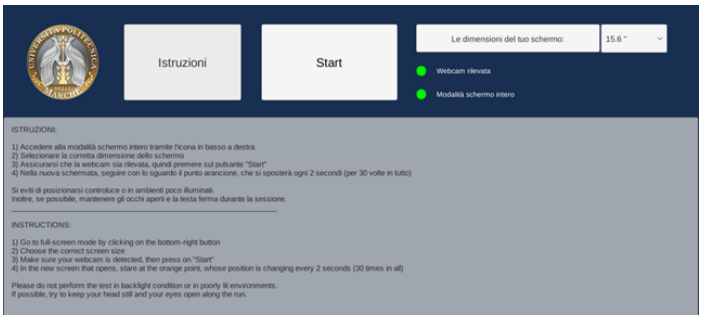

Figure 4: Main screen of the web application for collecting training pictures; by pressing on "Start", a new empty screen appears, in which the dots are shown in succession

For this reason, on the application interface, the user is asked to choose its own screen size, among the ones displayed in a dropdown list, ranging from 13 to 30 inches (i.e., the most common laptop and desktop sizes). By knowing the physical size of the screen and the display resolution (that can be automatically determined instead), it has been possible to express the dots coordinates in cm. At the end, 54 subjects agreed to participate, so the training dataset has been composed of 1620 photos.

The CNN for gaze tracking was implemented and trained using *python* programming language, running *Keras* API upon *TensorFlow* library support and with GPU acceleration.

Inputs of the CNN are the cropped face image (sized 224x224), the two cropped eyes images (sized 224x224) and a 1x4 face grid vector, expressing the portion of the entire image occupied by the face; in order to normalize the dataset, both the face and the eyes images are converted to grayscale, their contrast being increased by means of histogram normalization. The output is a two-dimensional vector containing x and y coordinates of the estimated display point (in centimeters). Faces and eyes are detected and cropped using the dlib [35] frontal face detector and 68 landmarks predictor, respectively. When a face box is found, the screen coordinates of its top left corner and its width and height are stored in the face grid vector; in this way implicit information can be retrieved about the user-screen relative pose. Adadelta [36] has been chosen among the optimizers implemented in *Keras*.

A python script was also realized for visualizing the scan path of the eyes over a displayed image; when launching it, a preset image is shown in full screen, while the webcam repeatedly takes pictures of the user and feeds the network with them. All the coordinates are stored and, when the image is closed, they are automatically processed in a manner that near points are clustered together, thus separating fixations from saccades [37]. Eventually, the colored clusters are depicted over the displayed image, together with straight lines connecting the subsequent ones, to give an idea on the path which the eyes followed overall.

4. Results

The proposed emotion and gaze detection systems were tested separately.

4.1. Evaluation of emotion detection performance

The test accuracy of each emotion category on the confusion matrix plotted as heatmap on Figure 5, shows some interesting results. The images with fear tag misclassified as surprise and disgust tag misclassified as anger has over 20% rate, these are results that are very often found in literature, mostly because these facial expressions are difficult to differentiate even for a human being, so they are often misinterpreted also in the dataset labeling phase. Moreover, as it can be seen, the emotion categories with more training samples resulting in higher accuracy compared to the ones with fewer training samples. Consequently, it is believed that a more evenly distributed dataset may improve the model accuracy further.

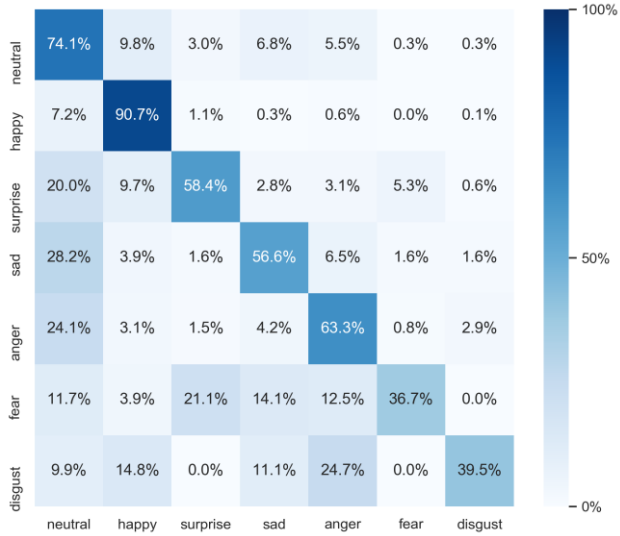

Figure 5: Accuracy of each emotion category

The model VGG13 we trained is also evaluated on EmotioNet [21] 2018 challenge dataset: the Ohio State University, on their website [38], has in fact made available their dataset to give anyone the opportunity to compare their results with those of the challenges of 2017 and 2018. The table shows the evaluation result.

Table 2: Emotionet evaluation results

Categories	Accuracy	F1
happy	**0.9770**	**0.9799**
anger	0.75	0.1198
disgust	0.0128	0.0099
sad	0.5955	0.2888
surprise	0.7059	0.3944

In this case only five emotions have been evaluated: this is because available photos related to Fear emotions were not enough to establish a meaningful comparison. In addition, no photos were made available for the comparison of the Neutral emotion.

4.2. Evaluation of eye-gaze detection performance

To evaluate the performance of the prediction the same software implemented to gather training samples has been used. The human eye detected using dlib [35] landmarks is the input of the python script that crops the region of interests and predicts the coordinates. They are expressed in centimeters and take the top-left screen corner as origin. A total of 20 subjects agreed to take part in the test and for each participant the predictions have been made for 30 different screen positions.

Every time the test is repeated the real and predicted coordinates have been saved in a database. To calculate the error between the predicted point and the real one the following formula has been used:

$$e_m = \frac{\sum_{i=1}^{n} | x_{i\,real} - x_{i\,pred} |}{n}$$

where $x_{i\,real}$ is the real coordinate where the subject *i* is looking at, $x_{i\,pred}$ is the predicted coordinate, *n* represents the total number of participants, and e_m the mean error (in centimeters) that has been calculated for any screen position. The same formula was used to evaluate the error on the y coordinate.

At this stage, 30 values of e_m have been calculated., The mean errors have been aggregated in 12 more meaningful values, as no significant variations have been found in comparison with the high number of screen positions. Each of them represents the error in a specific screen area (see Figure 6 and 7).

Both the heatmaps in Figure 6 and 7 show a higher error value in the top side of the screen (up to 11 cm), but interesting results have been achieved for the other screen positions. In this case the error goes between 0.02cm to 7.2cm. As the heatmap shows, the model still needs some improvement since the accuracy strongly depends on the area the subject stares at, but this experiment proves that it is possible to reach very accurate gaze predictions.

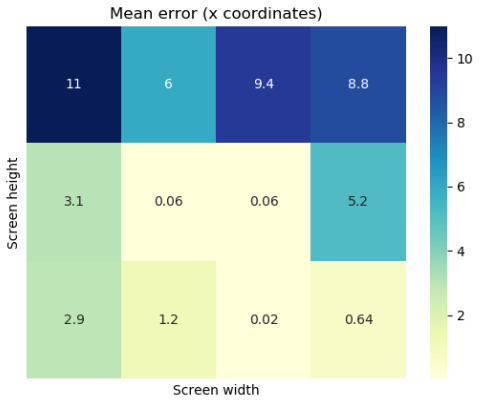

Figure 6: Mean error (cm) for each screen area for the x coordinate.

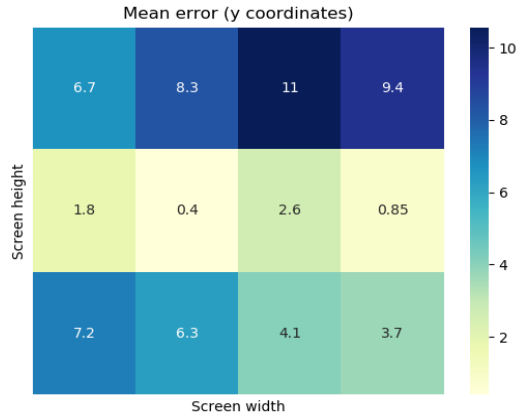

Figure 7: Mean error (cm) for each screen area for y coordinate.

5. Conclusion and discussion

This work proposed a system able to collect, data about user interactions/behavior with web applications potentially useful to support UX analysis. In particular, the proposed toolkit makes use of a deep learning-based platform to allow an "in the wild" data collection during user's interactions with a web application/platform just using a normal webcam. Future studies will be conducted to define an automatic system able to support the analysis of collected data, visualize UX outcomes through the provision of proper KPIs, and suggest useful design guidelines, based on the observed users' behavior, to improve UX design.

Regarding the emotion detection module, experimental results showed that the accuracies of facial expression categories such as fear and disgust are low mainly because of the small dataset of the corresponding category. The other facial expression categories however have reached relatively high classification accuracies. The evaluation results on EmotioNet dataset also supports the results of our experiment. Future work will be focused on the collection of more data on the small dataset categories to improve the classification accuracies of those corresponding categories

and on the exploration of generative adversarial networks to increase recognition accuracy.

About the gaze detection module, has been showed that the average error that can be reached is widely improvable and still does not reach the levels of accuracy of iTracker: this is due both to the small size of the dataset used for training and to the heterogeneity of the environments with which the user may be operating, in particular distance between user and screen, screen size, webcam position, etc. One of the future main works are undoubtedly the expansion of the dataset used to train the gaze network, unfortunately still very limited, adopting crowdsourcing tools.

Finally, data related to users' emotions and behavior, automatically collected through the proposed tool, can be used to improve recommendation systems based on collaborative filtering approaches. Current recommendation systems actually consider only data related to product features, customer preferences, customer demographic data (e.g., age, gender) historical data of purchases, and environmental factors (e.g., time, location). The availability of a huge amount of data related to what users observe the most, and what they feel while browsing e-commerce, opens new possibilities for customization and adaptation of recommendations, based on the real taste and interest of the user. Future studies should go in this direction.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The toolkit is part of the NextPerception project that has received funding from the European Union Horizon 2020, ECSEL-2019-2-RIA Joint Undertaking (Grant Agreement Number 876487). The authors would also thank the research fellow Lorenzo Marchesini for its support in the design and development of the eye gaze detection system.

References

[1] A. Talipu, A. Generosi, M. Mengoni, L Giraldi, "Evaluation of Deep Convolutional Neural Network architectures for Emotion Recognition in the Wild," in 2019 IEEE 23rd International Symposium on Consumer Technologies, 25-27, 2019. IEEE https://doi.org/10.1109/ISCE.2019.8900994

[2] C. Lallemand, G. Gronier, V. Koenig, "User experience: A concept without consensus? exploring practitioners' perspectives through an international survey," Computers in Human Behavior, **43**, 35–48, 2015. https://doi.org/10.1016/j.chb.2014.10.048

[3] J. Jabbar, I. Urooj, W. JunSheng, N. Azeem, "Real-time sentiment analysis on E-commerce application" in 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC) https://doi.org/10.1109/ICNSC.2019.8743331

[4] X. Fang, J. Zhan, "Sentiment analysis using product review data," Journal of Big Data, **2**(1), 5, 2015. https://doi.org/10.1186/s40537-015-0015-2

[5] S. Hedegaard, J. G. Simonsen, "Extracting usability and user experience information from online user reviews," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2013. https://doi.org/10.1145/2470654.2481286

[6] J. Nielsen, "The 90-9-1 rule for participation inequality in social media and online communities," 2016. Retrieved 20.07.2020, from https://www.nngroup.com/articles/participation-inequality/

[7] S. Jaiswal, S. Virmani, V. Sethi, K. De, P. P. Roy, "An intelligent recommendation system using gaze and emotion detection," Multimedia Tools and Applications, **78**(11), 14231-14250, 2019. https://doi.org/10.1007/s11042-018-6755-1

[8] I. Portugal, P. Alencar, D. Cowan, "The use of machine learning algorithms in recommender systems: A systematic review," Expert Systems with Applications, **97**, 205-227, 2018. https://doi.org/10.1016/j.eswa.2017.12.020

[9] A. Bielozorov, M. Bezbradica, M. Helfert, "The role of user emotions for content personalization in e-commerce: literature review," in International Conference on Human-Computer Interaction, 2019. https://doi.org/10.1007/978-3-030-22335-9_12

[10] A. Generosi, S. Ceccacci, M. Mengoni, "A deep learning-based system to track and analyze customer behavior in retail store," in 2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), 2018. https://doi.org/10.1109/ICCE-Berlin.2018.8576169

[11] A. Generosi, A. Altieri, S. Ceccacci, G. Foresi, A. Talipu, G. Turri, ... & L. Giraldi, "MoBeTrack: a toolkit to analyze user experience of mobile apps in the wild," in 2019 IEEE International Conference on Consumer Electronics (ICCE), 2019. https://doi.org/10.1109/ICCE.2019.8662020

[12] E. Ganglbauer, J. Schrammel, S. Deutsch, M. Tscheligi, "Applying Psychophysiological Methods for Measuring User Experience: Possibilities, Challenges and Feasibility," in Workshop on User Experience Evaluation Methods in Product Development, August 25, 2009. Uppsala, Sweden.

[13] S. Hedegaard, J. G. Simonsen, "Extracting usability and user experience information from online user reviews," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2013. https://doi.org/10.1145/2470654.2481286

[14] V. Georges, F. Courtemanche, S. Senecal, T. Baccino, M. Fredette, P. M. Leger, "UX heatmaps: mapping user experience on visual interfaces," in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016. https://doi.org/10.1145/2858036.2858271

[15] R. Y. da Silva Franco, R. Santos do Amor Divino Lima, M. Paixão, C. G. Resque dos Santos, B. Serique Meiguins, "UXmood—A Sentiment Analysis and Information Visualization Tool to Support the Evaluation of Usability and User Experience," Information, **10**(12), 366, 2019. https://doi.org/10.3390/info10120366

[16] A. Altieri, S. Ceccacci, M. Mengoni, "Emotion-Aware Ambient Intelligence: Changing Smart Environment Interaction Paradigms Through Affective Computing," in International Conference on Human-Computer Interaction, 2019. Springer, Cham. https://doi.org/10.1007/978-3-030-21935-2_20

[17] A. Kartali, M. Roglić, M. Barjaktarović, M. Đurić-Jovičić, M. M. Janković, "Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches," in 2018 14th Symposium on Neural Networks and Applications(NEUREL), 2018. https://doi.org/10.1109/NEUREL.2018.8587011

[18] P. Ekman, V. F. Wallace. *Manual for the facial action coding system*. Consulting Psychologists Press, 1978.

[19] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," 1–25, 2018. https://doi.org/10.1109/TAFFC.2020.2981446

[20] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution" 2016. https://doi.org/10.1145/2993148.2993165

[21] C. F. Benitez-Quiroz, R. Srinivasan, Q. Feng, Y. Wang, A. M. Martinez, "EmotioNet Challenge: Recognition of facial expressions of emotion in the wild," 2017.

[22] D. W. Hansen, Q. Ji, "In the eye of the beholder: a survey of models for eyes and gaze," IEEE Trans Pattern Anal Mach Intell, **32**(3), 478–500, 2010. https://doi.org/10.1109/TPAMI.2009.30

[23] J. Sigut, S. A. Sidha, "Iris center corneal reflection method for gaze tracking using visible light," IEEE Transactions on Biomedical Engineering, **58**(2), 411-419, 2010. https://doi.org/10.1109/TBME.2010.2087330

[24] R. Valenti, T. Gevers, "Accurate eye center location and tracking using isophote curvature," in 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008. https://doi.org/10.1109/CVPR.2008.4587529

[25] H. Wu, Y. Kitagawa, T. Wada, T. Kato, Q. Chen, "Tracking iris contour with a 3D eye-model for gaze estimation," in Asian Conference on Computer Vision, 2007. https://doi.org/10.1007/978-3-540-76386-4_65

[26] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, "Appearance-based gaze estimation in the wild," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019. https://doi.org/10.1109/CVPR.2015.7299081

[27] S. Winkler, R. Subramanian, "Overview of eye tracking datasets," in 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX), 2019. https://doi.org/10.1109/QoMEX.2013.6603239

[28] J. Lemley, A. Kar, A. Drimbarean, P. Corcoran, "Convolutional neural network implementation for eye-gaze estimation on low-quality consumer imaging systems," IEEE Transactions on Consumer Electronics, **65**(2), 179-187, 2019. https://doi.org/10.1109/TCE.2019.2899869

[29] X. Zhang, Y Sugano, M. Fritz, A. Bulling, A., "It's written all over your face: Full-face appearance-based gaze estimation," in Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition Workshops, 2017. https://doi.org/10.1109/CVPRW.2017.284
[30] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, A. Torralba, "Eye tracking for everyone," in Proceedings of the IEEE conference on computer vision and pattern recognition 2016. https://doi.org/10.1109/CVPR.2016.239
[31] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012. https://doi.org/10.1145/3065386
[32] 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, 2010. https://doi.org/10.1109/CVPRW.2010.5543262
[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition 2016. https://doi.org/10.1109/CVPR.2016.308
[34] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[35] D. E. King, "Dlib-ml: A machine learning toolkit." The Journal of Machine Learning Research, **10**, 1755-1758, 2009.
[36] M. D. Zeiler, "Adadelta: an adaptive learning rate method," arXiv preprint arXiv:1212.5701, 2012.
[37] L. Chamberlain, "Eye tracking methodology; Theory and practice," Qualitative Market Research: An International Journal, 2007. https://doi.org/10.1108/13522750710740862
[38] EmotioNet Challenge, Retrieved July 30, 2020 from https://cbcsl.ece.ohio-state.edu/EmotionNetChallenge/index.html