Volume XV Number 3, 2023

UX and Machine Learning – Preprocessing of Audiovisual Data Using Computer Vision to Recognize UI Elements

Martin Čejka[2], Jan Masner[1], Jan Jarolímek[1], Petr Benda[1], Michal Prokop[3], Pavel Šimek[1], Petr Šimek[1]

[1] Department of Information Technologies, Faculty of Economics and Management, Czech University of Life Sciences Prague, Czech Republic

[2] Department of Information Engineering, Faculty of Economics and Management, Czech University of Life Sciences Prague, Czech Republic

[3] Department of Management, Faculty of Economics and Management, Czech University of Life Sciences Prague, Czech Republic

Abstract

This study explores the convergence of user experience (UX) and machine learning, particularly employing computer vision techniques to preprocess audiovisual data to detect user interface (UI) elements. With an emphasis on usability testing, the study introduces a novel approach for recognizing changes in UI screens within video recordings. The methodology involves a sequence of steps, including form prototype creation, laboratory experiments, data analysis, and computer vision tasks. The future aim is to automate the evaluation of user behavior during UX testing. This innovative approach is relevant to the agricultural domain, where specialized applications for precision agriculture, subsidy requests, and production reporting demand streamlined usability. The research introduces a frame extraction algorithm that identifies screen changes by analyzing pixel differences between consecutive frames. Additionally, the study employs YOLOv7, an efficient object detection model, to identify UI elements within the video frames. Results showcase successful screen change detection with minimal false negatives and acceptable false positives, showcasing the potential for enhanced automation in UX testing. The study's implications lie in simplifying analysis processes, enhancing insights for design decisions, and fostering user-centric advancements in diverse sectors, including precision agriculture.

Keywords

Usability, UX, audiovisual data, computer vision, frame extraction, object detection, YOLOv7, precision agriculture.

Čejka, M., Masner, J., Jarolímek, J., Benda, P., Prokop, M., Šimek, P. and Šimek, P. (2023) "UX and Machine Learning – Preprocessing of Audiovisual Data Using Computer Vision to Recognize UI Elements", *AGRIS on-line Papers in Economics and Informatics*, Vol. 15, No. 3, pp. 35-44. ISSN 1804-1930. DOI 10.7160/aol.2023.150304.

Introduction

The agricultural industry exhibits a unique set of characteristics that sets it apart from other sectors. In particular, the availability of funding for research and development is limited compared to other industries. Consequently, implementing user experience (UX) testing practices in the agricultural sector must rely heavily on automated approaches in the future. The limited financial resources within the industry necessitate such automation to obtain necessary insights and data to enhance agricultural practices.

During UX testing, it is now possible through recording technologies to store audiovisual data of the participant's movement within the user interface (UI) and the participant's eye movement (using eye tracking technology) or other data, including biometric data. Obtaining these types of data (i.e., audiovisual data) using dedicated UX laboratories is the first step in developing machine learning technologies that could allow UX experts to process and evaluate the data automatically and efficiently. The reasoning is that the information about users' behavior is stored within these data. The new technology should automatically detect this behavior (or deviations from the expected behavior), thus making the data evaluation process faster and more efficient.

Although there are many studies on UX evaluation, few works have addressed the optimization and automation of this process (Aviz et al., 2019).

Since it is not yet possible to analyze this data algorithmically, it is necessary to process it manually (usually by UX experts), which, however, leads to extreme time and thus financial complexity of the analysis, especially in the case of many tested users (Harms, 2019). Experts often find this repetitive work tedious. Using human labor to explain and measure UX is inefficient (Koonsanit and Nishiuchi, 2021). A method based on Computer Vision could be a promising way to tackle the automatic evaluation of UX testing data. Computer vision techniques involve analyzing and understanding digital images and videos with the help of algorithms and mathematical models. (Batch et al., 2023) propose UXSense – a method for extracting multi-modal features of human behavior from video and audio footage using machine learning (ML) to support UX and usability professionals in their analysis of user session data using interactive visualization. Furthermore, computer vision in video analysis has gained tremendous attention in recent years. Video analysis involves extracting meaningful information from a sequence of frames. In contemporary times, computer vision plays a significant role in enhancing the field of precision agriculture, commonly referred to as agro-vision. Its applications encompass various tasks, including the monitoring and characterization of crops, weed management, assistance in harvesting, guiding agricultural vehicles, and creating yield maps. (Patrício and Rieder, 2018; Mavridou et al., 2019; Wang, Zhang and Wei, 2019; Bulanon et al., 2020)

To successfully develop a computer vision-based method, it is, therefore, necessary first to create a frame extraction algorithm that would consider only distinct frames of the record. (Harrison, Beverly L. and Baecker and Ronald, 1992) pointed out years ago that user pressure on the quality of applications in terms of usability and UX is constantly increasing. Therefore, the potential results of the present research may also be demanded at the commercial level. The development of this type of automation should optimize the analysis process, where only relevant data relevant to the research is presented in the output from the entire testing record without wasting time and risking overlooking essential facts. Moreover, as concluded by (Novák et al., 2023), the evaluation automation of UX and usability is a current research trend, even though not much addressed by the researchers yet.

This study was conducted within a project aimed at developing and optimizing methods that would, in the future, enable automatic evaluation of audiovisual data from usability and UX testing using AI-based methods. These methods can be used to develop applications specifically for the agricultural sector. In this study, we focus on a possible prediction of user actions associated with common form elements (text fields, select boxes, dropdown lists).

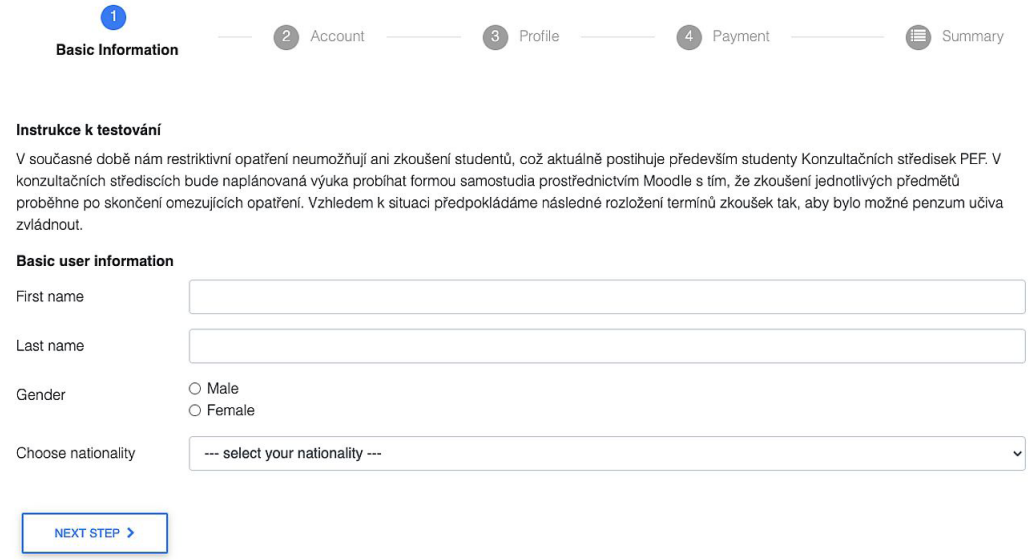
The main objective of this paper is to explore the use of computer vision methods for detecting UI objects within a video stream during usability tests conducted in a laboratory setting. The primary focus of this study is to assess the significance of changes between consecutive frames in the video recording. The aim is to identify pertinent images for the computer vision task. Subsequently, the findings from this task can be integrated with eye-tracking data to pinpoint the UI elements users target at specific moments.

Materials and methods

In the introductory section, the significance of the frame extraction algorithm within the realm of automated video processing was highlighted. This entails the partitioning of a digital image or video into distinct segments or regions guided by their inherent attributes. The research methodology comprises a sequence of steps, commencing with creating an experimental prototype form (step 1). Subsequently, controlled experiments were conducted in a laboratory setting to collect audiovisual data (step 2). In step 3, the data is subjected to computational analysis to extract unique frames (step in the form). Lastly, we investigated the possibility of using computer vision methods to identify the UI elements (step 4).

Experimental prototype

Considering technology solutions to the specific needs and challenges of the agrarian sector, a basic web form was created. By creating an application that mirrors the digital tools commonly employed in the agrarian sector - such as precision agriculture applications, applications for subsidies, and supply chain management interfaces—we can thoroughly follow the common environment within this specialized technological landscape. The application was designed to mimic a stepwise process, as depicted in Figure 1. The design was developed using the Bootstrap 4 CSS framework, which provides a cohesive and consistent appearance across numerous form designs utilized on the web. The web page layout incorporated several form elements, which are enumerated in Table 1.



Source: own procesing

Figure 1: Example of the web page prototype application form.

Input name	HTML Element	Label alignment	Prototype step
Simple text input	<input type="text">	inline	1
Number	<input type="number">	inline	1
Radio buttons 2 choices	<input type="radio">	inline	1
Radio buttons	<input type="radio">	separately	2,3
Checkboxes	<input type="checkbox">	separately	2,3
Single checkboxes	<input type="checkbox">	inline	4
Select box	<select>	inline	1,3
File upload	<input type="file">	inline, separately	1,3
Textarea	<textarea>	separately	2,3

Source: own procesing

Table 1: Description of the elements of the created form for the experiment.

Experimental protocol

The experiments were held in the Laboratory of usability (under the Human Behavior Research Unit; HUBRU) at the Czech University of Life Sciences Prague. The laboratory is composed of two separate rooms. The test room is a soundproof space containing chairs, computers, and an eye-tracking device as shown in Figure 2. The control room, which to the primary room and is used for the supervision of the research activities. The moderators have the option of either monitoring the participants from the control room or being physically present in the primary room to communicate with them. There are three cameras and ambient microphones installed in the primary room, and communication between the two rooms is feasible in either direction if required. The technologies used for data collection are described in Table 2.

Source: own procesing

Figure 2: Laboratory of usability (HUBRU) used to conduct the experiments.

Device	Description
PC	Windows 10 Professional edition
Screen	Full HD (1920x1024)
24-inch display	
Peripherals	Standard mouse and Keyboard
Web browser	Google Chrome
Eye tracker	Tobii Pro X2-60 (60hz; see Figure 3)

Source: own procesing

Table 2: Technologies used for data collection.

The eye tracker was tuned with the default parameters for the whole experiment. At the beginning of each testing session, participants were introduced to the scope and purpose of the experiment. After that, they were instructed to find a comfortable seating position. Participants received instructions about the testing method from the moderator. Then, a nine-point calibration of the eye tracker was carried out. Each participant had to participate in a preliminary task to acquire familiarity with the equipment.

In the following step, the participants followed the prototype application according to instructions. The task was finished by form submission. The instructions were the following:

- Switch to the web browser window.
- Read the introductory paragraph.
- Fill out the following form. Consider it as a registration to a web service.
- Fill in all the following steps and click next until the summary section.
- Click the submit button to finish.

During the experiment, the audiovisual data were acquired by the Tobii Pro Studio software, which operates the eye tracker. The collected data was then filtered and analyzed by a UX expert to select appropriate samples for subsequent data processing using computer vision techniques.

Frame extraction method

This task aimed to automate identifying screen changes in eye-tracking videos using computer vision algorithms. A screen change can be naturally understood as a major change in pixel values. In our case, we aim to identify moments when a user moves to the next page while filling out the presented form. Video records were loaded and cut into frame images and were converted to grayscale to reduce potential error sources. The structural similarity index (SSIM) in Formula 1, as well as the mean squared error (MSE) in Formula 2, were computed on every two consecutive frames of the example video. MSE expresses error signals as the difference between the original and distorted signals. When comparing images, the MSE – while simple to implement – is not highly indicative of perceived similarity. SSIM aims to address this shortcoming by taking texture into account (Wang et al., 2004; Wang and Bovik, 2009)

$$S(x,y) = l(x,y) \cdot c(x,y) \cdot s(x,y) = \\ = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right) \cdot \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right) \quad (1)$$

Formula 1 - SSIM, where μ_x and μ_y are (respectively) the local sample means of x and y, σ_x and σ_y are (respectively) the local sample standard deviations of x and y, and σ_{xy} is the sample cross-correlation of x and y after removing their means. The items C_1, C_2, and C_3 are small positive constants that stabilize each term, so that near-zero sample means, variances, or correlations do not lead to numerical instability.

$$MSE(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \quad (2)$$

Formula 2 - Mean Squere Error (MSE), where x and y represents pixel values of the two executive frames and N is number of pixels.

We implemented a floating threshold dependent on SSIM results calculated as the mean SSIM score values in addition to constant c, as shown in Formula 3. Frames with SSIM scores lower than the threshold are considered dissimilar to the next consecutive image. They, therefore, are extracted together with their timestamp - calculated as a multiplication of image order index and frame rate, as shown in Formula 4. Extracted frames with timestamps are suspected of representing major screen changes. Coming from (Shultz et al., 2011) we then evaluated our algorithm using confusion matrix and its related metrics as shown in Table 3.

$$\theta = \frac{\sum_{i=0}^{n} s_i}{n} + c \quad (3)$$

Formula 3 – Floating threshold calculation dependent on mean value of SSIM scores s_i and constanc c that is manually set to -0.03.

$$t_d = k_i \cdot v \quad (4)$$

Formula 4 - Frame to timestamp conversion, k_i is frame order number and v is frame rate being ~0.067.

Sensitivity / Recall	$TPR=\frac{TP}{TP+FN}$
Precision	$PPV=\frac{TP}{TP+FP}$
Accuracy	$ACC=\frac{TP+TN}{TP+FP+TN+FN}$
F1 score	$F1=\frac{2TP}{2TP+FP+FN}$

Source: own procesing

Table 3: Evaluation metricss of our computer vision experiments are sensitivity, precision, accuracy and F1 score. Confusion matrix contains true positives (TP), true negatives (TN), false positives (FP), false negatives (FN).

Identification of UI elements

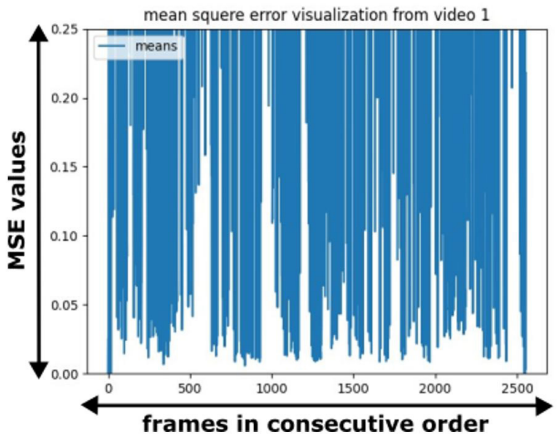
The objective of this step is to detect elements of the user interface, such as buttons, checkboxes, date pickers, file dialog, payment menu, radio button, select box, textbox. Identification (getting screen coordinates) of those items using object detection would empower video processing automation and bring new opportunities for video analysis.

The utilization of YOLOv7 for object detection is well-justified due to its exceptional performance and efficiency in real-time image analysis. YOLOv7's ability to balance accuracy and speed makes it an ideal choice for applications requiring swift and reliable object detection.

We collected and annotated various images of screenshots containing UI elements for object detection training. We divided dataset into a training set, validation set and training set. We applied augmentation steps – flip and 90° rotation to extend training set. Images were stretched into maxiumum allowed shape for YOLOv7. All images were converted to grayscale. We used mplementation of paper YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. (Wang, et al., 2022) on a pretrained COCO dataset model YOLOv7 (Lin et al., 2014). We then evaluated using confusion matrix and related metrics.

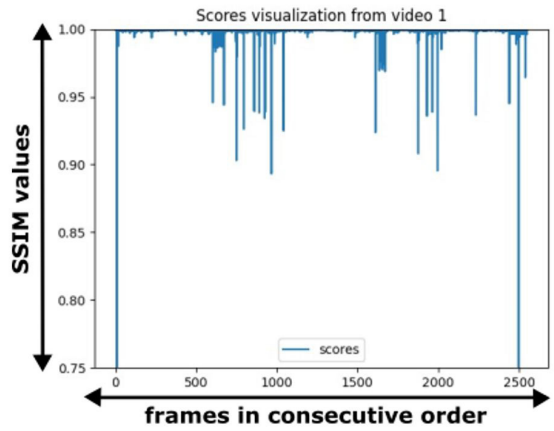
Results and discussion

In Figures 3 and 4, we show an example of both MSE and SSIM applied on our eye tracking video records and visualized in time. The example shows that SSIM can identify screen changes while MSE is not capable of detecting them correctly. Choosing SSIM we processed thirty videos according to technical details of the run introduced in Table 4. The results of evaluating the detection of the unique form step are shown in a confusion matrix in Table 5. To further evaluate the results, we calculated various metrics, as shown in Table 6. All screen changes have been detected. We achieved zero false negatives and, therefore, high sensitivity and accuracy values. We obtained high true negatives number 111 630. This result can be attributed to the prevailing inactivity within the video content, with screen changes occurring in only approximately 0.7% of the total video duration. We reached 630 false positives, bringing precision down to 0.16 and F1 score to 0.364. But exploring false positives showed that the algorithm generates only four types them. In fact, it worked correctly as the screen did change in all cases, although not from one form to another, which was the intended output. In Figures 5, 6, 7, and 8, we may see detections of the opening select box, file browser dialog, change of content in file browser dialog, and switching of the eye tracking at the end of the session. The last-named error can be eliminated by either excluding all black screens or ignoring the last screen change of the video if the timestamp is just before the end of the record. From the nature of the other three types of errors, we stated that they all represent a screen change (although the user didn't move to the next screen) and cannot be worked around by a simple algorithm. Considering this, we did detect all screen changes with 100%. accuracy (False Negative is 0, and False Positive would be 0 if the above-described detected events are considered screen changes, which, in pixel comparison, they are).



Source: own procesing

Figure 3: Mean Square Error (MSE) visualization of an example video where each two consecutive frames are compared.



Source: own procesing

Figure 4: Structural Similarity Index (SSIM) visualization of an example video where each two consecutive frames are compared.

Environment	CULS AI-LAB (Nvidia P40)
Number of processed videos	30
Average video size	1.5 GB
Average video length	4,5 min
Total computation time	19h 47m
Number of compared frame pairs	112 440
Detected screen changes candidates	790 (26.3 per video in average)
Threshold constant c	-0.03

Source: own procesing

Table 4: Technical details of the run points out on time inefficiency of the proposed solution.

Predicted class Ground truth class	Positive (1)	Negative (0)
Positive (1)	TP 180	TN 111 630
Negative (0)	FP 630 / * 0	FN 0

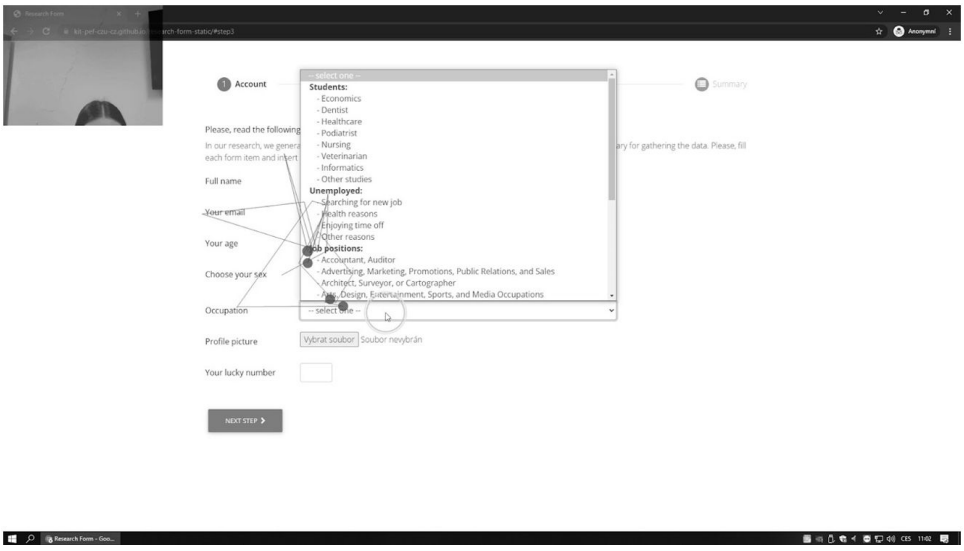
Source: own procesing

Table 5: Confusion matrix of the screen change detection using SSIM. Shows unbalanced distribution of data between tru positives and true negatives. * All 630 false positives might be concluded not to be a failure of the algorithm if error types in Figures 5,6,7,8 are considered not errors.

Sensitivity	1.0
Precision	0.22 / * 1
Accuracy	0.998 / * 1
F1 score	0.364 / * 1

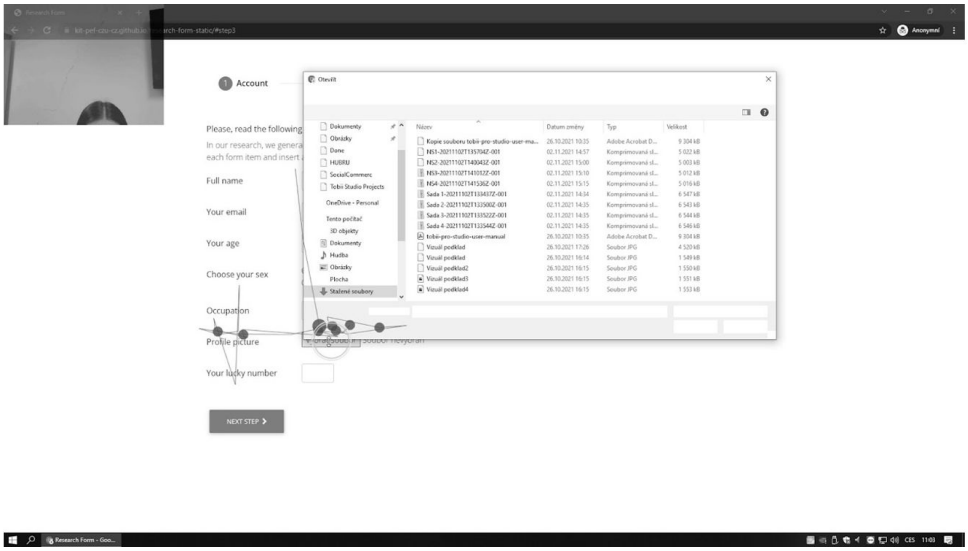
Source: own procesing

Table 6: Related metrics of the screen change detection using SSIM. * All metrics might be concluded equal to 1 if error types in Figures 5,6,7,8 are considered not errors.



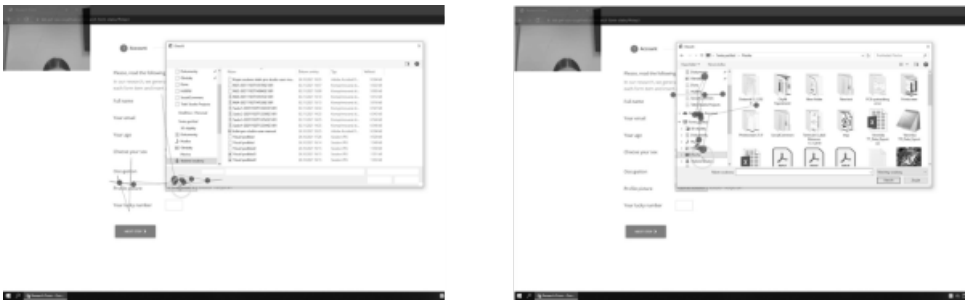
Source: own procesing

Figure 5: Error of type 1 - user opens a selectbox component which triggers higher SSIM value and is considered a screen change.



Source: own procesing

Figure 6: Error of type 2 – user opens a file browser dialog component which triggers higher SSIM value and is considered a screen change.



Source: own procesing

Figure 7: Error of type 3 – user changes content inside the selectbox and algorithm chooses a pair to be a screen change.

Source: own procesing

Figure 8: Error of type 4 - As the recording session ends, it detects a screen change at the end of the video.

The situation with known errors that cannot be worked around, and the very high computation cost clearly calls for a machine learning based object detection algorithm to do the rest of the work. As object detection would also empower both process automation and new opportunities for video analysis. YOLOv7 was trained using attributes in Table 7 and evaluated manually resulting in confusion matrix in Table 8 and related metrics in Table 9. Considering the findings presented, we discern that there exists significant promise for enhancements in several key facets of our approach. These potential areas of improvement include the fine-tuning process, the expansion of our dataset, the split of frames into smaller frames rather than applying stretching techniques,

and the optimization of dataset distribution to ensure an sufficient representation of each class.

Environment	Google Colab (Tesla T4)
Training set examples	435 (70%)
Validation set examples	43 (20%)
Test set examples	21 (10%)
Epochs	55
Batch size	16
Duration (minutes)	30
Image shape	608 x 608
Pretrained model	COCO

Source: own procesing

Table 7: Technical details of YOLOv7 train run – dataset split details and training parameters.

Confusion matrix		
Predicted class Ground truth class	Positive (1)	Negative (0)
Positive (1)	TP 107	TN does not apply
Negative (0)	FP 37	FN 9

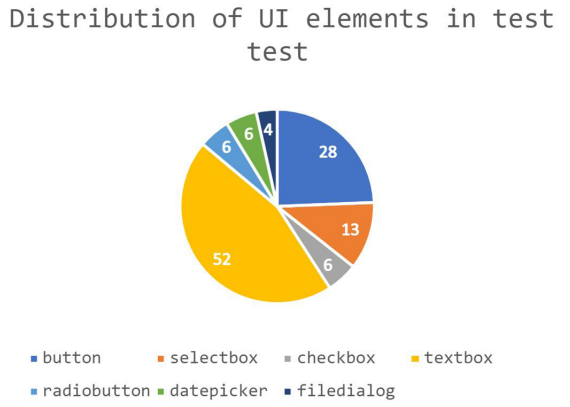
Source: own procesing

Table 8: Confusion matrix of results from our YOLOv7 implementation shows potential of object detection task as well as the need of improving both dataset and training parameters.

Sensitivity	0.922
Precision	0.743

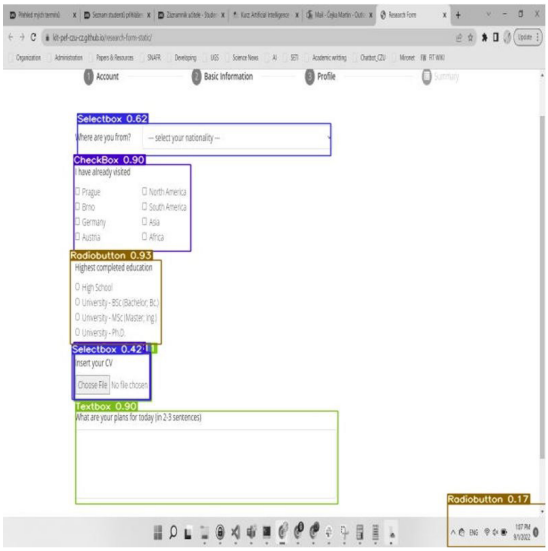
Source: own procesing

Table 9: While YOLOv7 resulting sensitiity is quiet high, precision metric shows space for improvement.



Source: own procesing

Figure 9: Graph shows unbalanced distribution of UI elements in test set.



Source: own procesing

Figure 10: Example from test set shows both great potential and great challenges in UI elements object detection.

The landscape of object detection remains dynamic, with diverse models continually emerging (such as a fresh YOLOv8). This study is a proof-of-concept solution and demonstrates feasibility while acknowledging the evolving nature of the field. Future strides are expected to introduce models with improved performance and adaptability. The presented solution is a foundational step, but the ongoing evolution of technology and research will likely yield better performances and models trained on broader data sets. This proof-of-concept highlights the vast potential for innovation in automation of UX testing evaluation.

Conclusion

In conclusion, computer vision techniques, specifically object detection, play a key role in the development of technologies for the automatic evaluation of audiovisual data from UX testing. The ability to automatically identify objects and regions in video data is the first step towards developing technology that can provide valuable insights into user behavior and enable researchers to make more informed design decisions. Recent advances in single-stage detectors push the boundaries of performance up and opens new opportunities for UX researchers to quickly and efficiently understand user behavior.

The resulting solution could significantly contribute to the automation of UX testing within the agricultural sector, encompassing various

applications in precision agriculture. These applications include the configuration and control of machinery and IoT devices, processes related to subsidy applications, and production reporting for subsidy claims. Given the generally lower ICT literacy in the agricultural sector, these applications often tend to be complex and challenging to navigate. The proposed approach could thus represent a substantial advancement in simplifying and automating UX testing in these applications, ultimately enhancing their user-friendliness and overall effectiveness.

Our future research is directed towards developing the aforementioned technology. Leveraging the video image segmentation algorithm as a foundational step, as expounded upon in this study. Following this, our investigation will utilize object detection, integrating with eye-tracking data to facilitate a comprehensive analysis aimed at delineating the precise points (i.e., UI elements) of user focus. This multi-faceted approach holds promise for advancing our understanding of user interactions within the studied context.

Acknowledgments

The results and knowledge included here have been obtained owing to support from the following grants – Internal grant agency of the Faculty of Economics and Management, Czech University of Life Sciences Prague, grant no. 2022A0015.

Corresponding author
Ing. Jan Masner, Ph.D.
Department of Information Technologies, Faculty of Economics and Management
Czech University of Life Sciences Prague, Kamýcká 129, 165 00, Prague, Czech Republic
E-mail: masner@pef.czu.cz

Martin Čejka ORCID No. 0000-0002-2909-486X
Jan Masner ORCID No. 0000-0003-4593-23061
Jan Jarolímek ORCID No. 0000-0001-7194-3055
Petr Benda ORCID No. 0000-0001-9651-8258
Pavel Šimek ORCID No. 0000-0001-9244-1476

References

[1] Aviz, I. L., Souza, K. E., Ribeiro, E., de Mello Junior, H. and da R. Seruffo, M. C. (2019) "Comparative study of user experience evaluation techniques based on mouse and gaze tracking", *WebMedia '19: Proceedings of the 25th Brazillian Symposium on Multimedia and the Web,* New York, NY, USA: ACM, pp. 53-56. ISBN 978-1-4503-6763-9. DOI 10.1145/3323503.3360623.

[2] Batch, A., Ji, Y, Fan, M., Zhao, J. and Elmqvist, N. (2023) "uxSense: Supporting User Experience Analysis with Visualization and Computer Vision", *IEEE Transactions on Visualization and Computer Graphics*, pp. 1-15. ISSN 1077-2626. DOI 10.1109/TVCG.2023.3241581.

[3] Bulanon, D. M., Hestand, T., Nogales, C., Allen, B. and Colwell, J. (2020) "Machine Vision System for Orchard Management", In: Sergiyenko, O., Flores-Fuentes, W., Mercorelli, P. (eds) "*Machine Vision and Navigation*", Springer, Cham., pp. 197-240. E-ISBN 978-3-030-22587-2. ISBN 978-3-030-22586-5. DOI 10.1007/978-3-030-22587-2_7.

[4] Harms, P. (2019) "Automated Usability Evaluation of Virtual Reality Applications", *ACM Transactions on Computer-Human Interaction*, Vol. 26, No. 3, pp. 1-36. ISSN 1073-0516. DOI 10.1145/3301423.

[5] Harrison, B. L. and Baecker R. M. (1992) "Designing Video Annotation and Analysis Systems", *Proceedings of the Conference on Graphics Interface '92*, Vancouver, British Columbia, Canada: Morgan Kaufmann Publishers Inc., pp. 157-166. ISBN 0969533810.

[6] Koonsanit, K. and Nishiuchi, N. (2021) "Predicting Final User Satisfaction Using Momentary UX Data and Machine Learning Techniques", *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 16, No. 7, pp. 3136-3156. ISSN 0718-1876. DOI 10.3390/jtaer16070171.

[7] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L. and Dollár, P. (2014) "Microsoft COCO: Common Objects in Context", *CoRR, abs/1405.0312*. [Online]. Available: http://arxiv.org/abs/1405.0312. [Accessed: May 17, 2023].

[8] Mavridou, E. , Vrochidou, E, Papakostas, G. A., Pachidis, T. and Kaburlasos, V. G. (2019) "Machine Vision Systems in Precision Agriculture for Crop Farming", J*ournal of Imaging*, Vol. 5, No. 12, 89 p. ISSN 2313-433X. DOI 10.3390/jimaging5120089.

[9] Novák, J. Š., Masner, J., Benda, P., Šimek, P. and Merunka, V. (2023) "Eye Tracking, Usability, and User Experience: A Systematic Review", *International Journal of Human–Computer Interaction*, pp. 1-17. ISSN 1044-7318. DOI 10.1080/10447318.2023.2221600.

[10] Patrício, D. I. and Rieder, R. (2018) "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review", *Computers and Electronics in Agriculture*, Vol. 153, pp. 69-81. ISSN 0168-1699. DOI 10.1016/j.compag.2018.08.001.

[11] Shultz, T. R. et al. (2011) "Confusion Matrix", in Sammut, C., Webb, G.I. (eds) "*Encyclopedia of Machine Learning".* Boston, MA: Springer US, pp. 209-209. E-ISBN 978-0-387-30164-8, ISBN 978-0-387-30768-8. DOI 10.1007/978-0-387-30164-8_157.

[12] Wang, A., Zhang, W. and Wei, X. (2019) "A review on weed detection using ground-based machine vision and image processing techniques", *Computers and Electronics in Agriculture*, Vol. 158, pp. 226-240. ISSN 0168-1699. DOI 10.1016/j.compag.2019.02.005.

[13] Wang, C.-Y., Bochkovskiy, A. and Liao, H.-Y. M. (2022) "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", *arXiv preprint arXiv:2207.02696* [Preprint]. DOI 10.1109/CVPR52729.2023.00721.

[14] Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. (2004) "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600-612. ISSN 1057-7149. DOI 10.1109/TIP.2003.819861.

[15] Wang, Z. and Bovik, A. C. (2009) "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures", *IEEE Signal Processing Magazine*, Vol. 26, No. 1, pp. 98-117. ISSN 1053-5888. DOI 10.1109/MSP.2008.930649.