Assessing Completeness in Training Data for Image-Based Analysis of Web User Interfaces

Sebastian Heil*
Technische Universitt Chemnitz
Chemnitz, Germany
sebastian.heil@informatik.tu-chemnitz.de
0000-0003-2761-9009

Maxim Bakaev*
Novosibirsk State Technical University
Novosibirsk, Russia
bakaev@corp.nstu.ru
0000-0002-1889-0692

Martin Gaedke
Technische Universitt Chemnitz
Chemnitz, Germany
martin.gaedke@informatik.tu-chemnitz.de
0000-0002-6729-2912

* Both authors contributed equally to the work.

Abstract

Analysis of user interfaces (UIs) based on their visual representation (screenshots) is gaining increasing popularity, institutionalizing the HCI vision field. Witnessing the same visual appearance of a UI like a human user provides the advantage of taking into account layouts, whitespace, graphical content, etc. independent of the concrete platform and framework used. However, visual analysis requires significant amounts of training data, particularly for the classifiers that identify UI elements and their types. In our paper we demonstrate how data completeness could be assessed in training datasets produced by crowdworkers, without the need to duplicate the extensive work. In the experimental session, 11 annotators labeled more than 42000 UI elements in nearly 500 web UI screenshots using the LabelImg tool with the pre-defined set of classes corresponding to visually identifiable web page element types. We identify metrics that can be automatically extracted for UI screenshots and construct regression models predicting the expected number of labeled elements in the screenshot. The results can be used in outlier analysis of crowdworkers in any existing microtasking platform.

Keywords: Machine Learning, Crowdworking, Image Recognition, Human-Computer Vision

In: S. Hölldobler, A. Malikov (eds.): Proceedings of the YSIP-3 Workshop, Stavropol and Arkhyz, Russian Federation, 17-09-2019–20-09-2019, published at http://ceur-ws.org

1 Introduction

Though futurologists have long been fearing that the development of AI is going to take jobs away from humans, it actually caused boom in the demand for microworking services. These involve the use of general human intelligence to complete tasks for which no algorithm is known or satisfactory efficient. The outcome is employed in either solving some practical problem, providing an online service, or improving an AI model based on machine learning. Popular examples include labeling images, audios and videos, moderating online content, sentiment analysis, translating short texts in other languages, etc. These tasks by and large involve unskilled and tedious work on data gathering and processing, so microworking services requestors can rarely get enough motivated volunteers and tend to rely on low-paid microworkers [1]. For instance, there are already disturbing reports about involuntary microservitude prisoners in Finland who have been assigned with data tagging jobs [2]. Aside from concerns what the AI taught by socially dubious teachers will be like, the consequence is the growing need for checking the quality of the outcome produced by such uninterested workers.

Quality is also the current focus for the crowdwork done via Internet (see review in [3]), and the number of specialized platforms has been growing lately: MTurk (2005), microworkers.com (2009), Yandex.Toloka (2014), Googles AutoML (2018), etc. Controlling completeness and accuracy of data used for training is particularly important, since these quality dimensions are linked to recall and precision in the resulting AI models. Todays trend is not just implementation of the output data quality assessment tools in the platforms: e.g. Yandex.Toloka allows specification of control rules for performance time, accuracy vs. the ground truth, majority consensus, etc. There is also a growing number of related meta-tools: CDAS [4], Crowd Truth [5], iCrowd [6], DOCS for MTurk [7], and more. Generally, they are concerned with online quality control and optimal assignment of tasks and mostly rely on performance in completed work, which is evaluated based on ground truth and majority consensus approaches. Nearly universally, these require that the same or very similar work is done by several workers, so that an accuracy measure could be calculated.

This work duplication is undesirable for some domains, where the tasks are labor-intensive and have no strictly correct outcome. Particularly, in our work we focus on user interface (UI) labeling the specification of UI elements positions and types in UIs visual representation. We propose assessing the output data completeness based on the expected number of objects that we predict for UI based on certain metrics that can be automatically calculated for the UI image. Most existing research even in general image labeling focuses on image complexity, in which the number of objects is only one of the dimensions. Particularly, image compression metrics, such as the popular JPEG or PNG, are known to be well correlated with image complexity, but their application for UIs assessment is specific and relatively novel (see in [8]). The potential advantage of the approach is increasing the efficiency of producing training data via removing the necessity to perform spare work to ensure the data quality.

The remainder of the paper is organized as follows. In Section 2, we detail the UI image-based (visual) analysis approach and describe the related software tool that we previously built. Further, we run experimental UI labeling session with 11 workers that processed about 500 screenshots of university website homepages. In Section 3, we analyze the collected data, present the characteristics of the dataset, and construct regression models for predicting the expected number of UI elements from JPEG, PNG, and entropy metrics combined with edge detection-based recognition. In the final section, we discuss the results, provide conclusions and outline directions for further research. So far, the greatest limitation of our work is lack of its testing to see if crowdworkers undermining data completeness can be identified in real conditions.

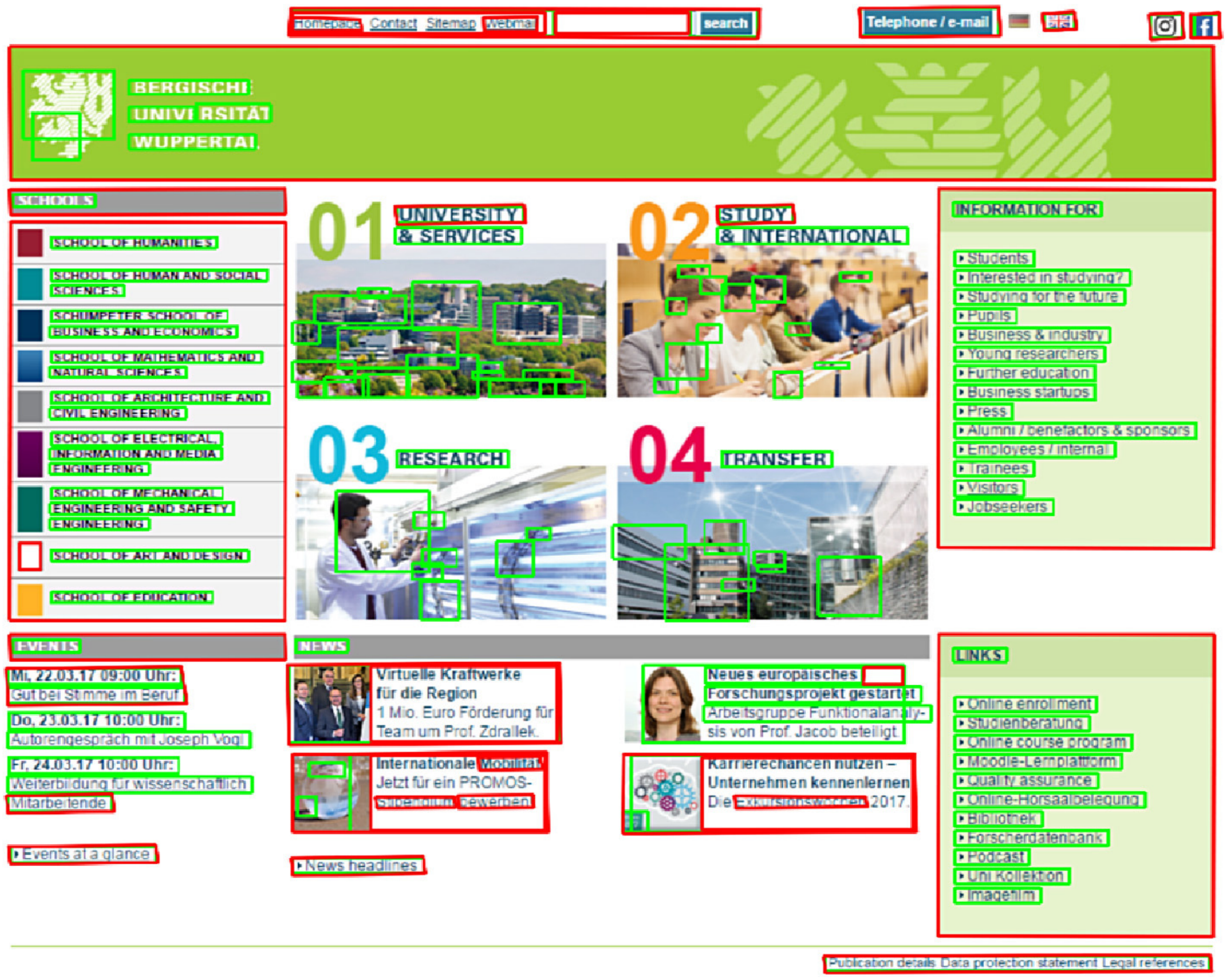
2 Methods

2.1 Web UI Visual Analysis

Image-based analysis of UIs is gaining in popularity, as it allows witnessing the same interface as the user, which is particularly important for web UIs. The drawback of this approach is that considerable amount of training data is needed (particularly for the classifiers that identify UI elements and their types), which is mostly produced through human UI labeling.

It is already widely noted that when data is annotated through crowdworking platforms, controlling its quality is of foremost importance [3]. Particularly for UI labeling tasks, data completeness can suffer if unfaithful crowdworkers optimize their task performance for better revenue / effort ratio. We believe that such outlier workers can be identified without adhering to duplicate labeling, on which ground truth and majority consensus are essentially founded. Objective characteristics of the material (that is, UIs being labeled) can provide meaningful clues on the degree of a worker performance's regularity.

Previously, we have developed a prototype visual analyzer tool capable of extracting several metrics from UIs visual representation [9]. It exhibited rather acceptable recognition of UI elements, which is mostly based on edge detection and identification of vertical and horizontal lines, rectangular forms, etc. (see in Fig.1). However, the performance of its trained classifiers responsible for detection of web UI elements types was found to be inadequate. As we are developing the enhanced visual analyzer, we are concerned with efficient collection of training data via web UIs labeling and assessment of its quality.



Homepage Contact Sitemap Webmail
search
Telephone / e-mail
BERGISCHE UNIVERSITÄT WUPPERTAL
SCHOOLS
SCHOOL OF HUMANITIES
SCHOOL OF HUMAN AND SOCIAL SCIENCES
SCHUMPETER SCHOOL OF BUSINESS AND ECONOMICS
SCHOOL OF MATHEMATICS AND NATURAL SCIENCES
SCHOOL OF ARCHITECTURE AND CIVIL ENGINEERING
SCHOOL OF ELECTRICAL, INFORMATION AND MEDIA ENGINEERING
SCHOOL OF MECHANICAL ENGINEERING AND SAFETY ENGINEERING
SCHOOL OF ART AND DESIGN
SCHOOL OF EDUCATION
01 UNIVERSITY & SERVICES
02 STUDY & INTERNATIONAL
03 RESEARCH
04 TRANSFER
INFORMATION FOR
Students
Interested in studying?
Studying for the future
Pupils
Business & industry
Young researchers
Further education
Business startups
Press
Alumni / benefactors & sponsors
Employees / internal
Trainees
Visitors
Jobseekers
EVENTS
Mi, 22.03.17 09:00 Uhr:
Gut bei Stimme im Beruf
Do, 23.03.17 10:00 Uhr:
Autorengespräch mit Joseph Vogl
Fr, 24.03.17 10:00 Uhr:
Weiterbildung für wissenschaftlich Mitarbeitende
Events at a glance
NEWS
Virtuelle Kraftwerke für die Region
1 Mio. Euro Förderung für Team um Prof. Zdrallek
Neues europäisches Forschungsprojekt gestartet
Arbeitsgruppe Funktionalanalysis von Prof. Jacob beteiligt
Internationale Mobilität
Jetzt für ein PROMOS-Stipendium bewerben
Karrierechancen nutzen – Unternehmen kennenlernen
Die Exkursionswochen 2017
News headlines
LINKS
Online enrollment
Studienberatung
Online course program
Moodle-Lernplattform
Quality assurance
Online-Hörsaalbelegung
Bibliothek
Forscherdatenbank
Podcast
Uni Kollektion
Imagefilm
Publication details Data protection statement Legal references

Figure 1: Recognition of UI elements in the visual analyzer tool.

2.2 UI Labeling and the Data Quality

Data completeness is an important attribute of overall data quality, which indicates comprehensiveness of available data with respect to a specific informational requirement. General crowdworking is arguably more often concerned with data accuracy, since tasks are rarely compound enough to be completed only partially. In UI labeling, completeness can be undermined if too few UI elements are identified, which would further lead to decreased recall of the automated visual analysis tool.

To assess data completeness in this domain, the ground truth and majority consensus approaches could be well used. That is, if a crowd worker repeatedly under-identifies UI elements, his or her output could be considered invalid, and no further tasks would be assigned. However, web UIs are currently very diverse and the full number of UI elements can vary dramatically. Processing a single UI takes considerable time, so each worker would only label a couple dozens of them, ruling out statistically meaningful comparison of averaged values per workers labeling different UIs. It means that in order for the ground truth and majority consensus approaches to be effective, several workers would have to process the same UI. I.e. the extensive labeling effort would have to be duplicated, without contributing much to the training data.

Instead, the expected number of elements in a web UI could be predicted without the involvement of human workers, based on metrics extracted from its image (screenshot, as demonstrated e.g. in [9]). Subsequently, a kind of outlier analysis [3] could be employed to identify workers whose performance consistently does not correspond to the expected values. To explore whether the prediction-based approach will hold true for the trusted dataset and to identify the significant metrics, we collected labeling data in an experimental session.

2.3 The Experiment Description

2.3.1 Participants

The workers in our study were student members of the Novosibirsk State Technical University (Russia) crowd-intelligence lab, who volunteered to work on the project. In total, there were 11 of them (6 male, 5 female), with age ranging from 20 to 24 (mean = 20.5, SD = 0.74), all Bachelor students of Applied Informatics major. All the workers had normal or corrected to normal vision and reasonable experience with web UIs and IT.

2.3.2 Material

The material was screenshots of higher educational organizations websites homepages (UIs). Initially, 10639 screenshots were collected automatically by the dedicated Python script crawling through URLs we took from various catalogues (DBPedia, etc.). The screenshots were made for full web pages, as they were rendered, not just of the part above the fold or of a fixed size. Then we hand-picked 497 screenshots from the population, using the following criteria:

1. University or college corporate website with reasonably robust functionality;
2. Not overly famous university;
3. Website content in English and reasonably diverse (i.e. no photos-only websites);
4. Reasonable diversity in website designs (colors, page layouts, etc.).

2.3.3 Design

The experiment used between-subjects design each UI screenshot was processed only by one worker, so there was no duplication of work. The independent and derived independent variables were:

1. The size of the UI screenshot file in PNG-24 format, in MB: PNG_filesize;
2. File size for the same screenshot in JPEG-100 format, in MB: JPEG_size;
3. The number of elements metric automatically produced by the visual analyzer for a UI screenshot: VA_Elements;
4. Entropy value obtained for the .png file through MATLABs entropy(I) function: M_Entropy.

The dependent variable in our study was the number of UI elements labeled in UI screenshot by worker: N_Elements.

2.3.4 Procedure

For labeling the UIs, the workers used LabelImg tool. It allows drawing bounding rectangle around an image element, specifying a label for it (choosing from the set of pre-defined classes or adding a custom class), and saving the results as XML files in PASCAL VOC format. The workers were provided with instruction on using the tool and were given the set of pre-defined classes specific for web UIs (see in Table 1).

The 497 UI screenshots were distributed between the student workers nearly equally and based on their alphabetical order (no random assignment). Each worker used his or her judgment in deciding which UI elements to label, but they were asked to achieve maximum completeness in each UI. In total, it took the workers 6 days to complete their assignment.

3 Results

3.1 Descriptive Statistics

The workers in total labeled 495 UI screenshots (2 erroneous ones were removed). This resulted in 42716 labeled UI elements, of which 39803 (93.2%) belonged to the pre-defined classes (shown inee Table 1). Example of a screenshot being labeled with the LabelImg tool is provided in Fig. 2.

Table 1: The pre-defined classes and the frequencies in the student workers results.

Class name	**Class description**	**Frequency**
Graphical content elements:		
image	foreground images that the web page displays	4046
background image	images that are used as background, i.e. other UI Elements are placed on top of them and they have no semantic meaning	673
panel	an area that is visually separated from its surroundings by borders, shadows, and/or background color and contains at least one other UI element	573
Textual content elements:		
list	any list (numbered or unnumbered) that uses bullet points, numberings, borders, background color etc. to display a set of similar items	305
table	any visually recognizable table (using alignment, lines or background color to represent rows and columns)	24
paragraph	a portion of text consisting of one or more lines of text that are not visually separated by white space and/or indentation from other text	1964
textblock	two or more subsequent paragraphs of text	728
text	any other portion of text that is neither a label nor a paragraph or textblock	7949
symbol	any graphical symbol, can appear on buttons, tabs, links, in texts etc. or separately	1803
Interface elements:		
checkbox	must be labeled one-by-one, without the accompanying text (which must be marked as label)	10
radiobutton	must be labeled one-by-one, without the accompanying text (which must be marked as label)	199
selectbox	a listbox that would expend when clicked, displaying several options which can be selected or multi-selected	863
textinput	single line (including password field, data/calendar, etc.)	375
textarea	multi line	69
button	if the button displays text on it, please additionally label the text of the button as type ”label” (see below)	2571
label	a small portion of text, typically one word or only few words, that are used together with another UI control like a radiobutton	3288
tabs	intra-page tabs created using HTML/CSS/JS, not browser tabs, please place the rectangle around the tab handle	427
scrollbar	both intra-page e.g. inside textareas and the main scrollbar of the entire page if displayed	59
pagination	should span the entire pagination controls area, typically the next and previous buttons and page links	92
link	can be inside text (hyperlink), in navigation, etc.	13785

Between the UIs, N_Elements per UI ranged from 12 to 230, mean = 86.3, SD = 38.3, RSD = 44.3%. The Kolmogorov-Smirnov normality test suggested that normality hypothesis had to be rejected for N_Elements

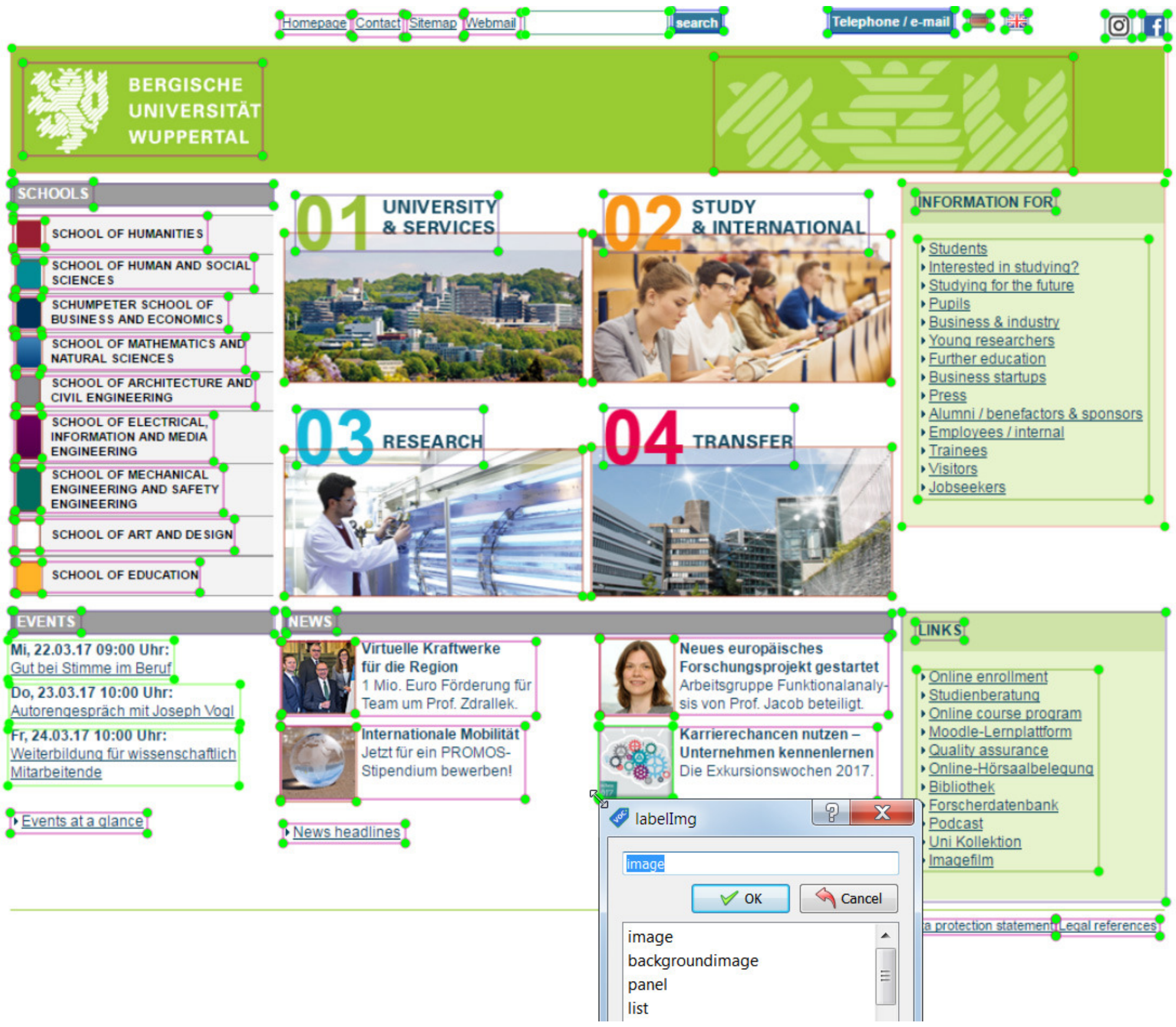


Figure 2: Example of a screenshot being labeled by a worker in LabelImg tool (PascalVOC).

($D_{495} = 0.047$, $p = 0.01$).

Between the workers, the average N_Elements per UI ranged from 44.5 to 121.6, mean = 86.4, SD = 22.3. Detailed statistics is provided in Table 2 (workers' names are shortened to initials). When counting the classes, obviously errorneous ones (e.g. butto) were removed from the consideration. Notably, the relative standard deviations (RSD), bar one outlier worker (SMl with RSD = 86.26%), ranged in a rather narrow interval of 24.24-49.05%. The Shapiro-Wilks test suggested that normality hypothesis could not be rejected ($W_{11} = 0.972$, $p = 0.903$).

3.2 Analyzing and Predicting the Number of UI Elements in Screenshots

Running the 495 UI screenshots processed by the workers through our visual analyzer software, we were able to obtain the number of UI elements metric (VA_Elements) for 440 of them (another 55 or 11.1% encountered technical problems). The resulting VA_Elements ranged from 4 to 278, mean = 65.4, SD = 32.7, RSD = 50.1%. Hence, on average human workers recognized 1.32 times more UI elements than the automation tool. The Kolmogorov-Smirnov test suggested that normality hypothesis had to be rejected for VA_Elements ($D_{440} = 0.105$, $p < 0.001$).

We found that Pearson correlation between N_Elements and VA_Elements per UI was highly significant ($r_{440} = 0.381$, $p < 0.001$). The correlations for JPEG_filesize ($r_{495} = 0.278$, $p < 0.001$), PNG_filesize ($r_{495} = 0.174$,

Table 2: Descriptive statistics for the labeled UI elements per workers.

Worker name (gender)	UIs labeled	UI elements	Classes used	Mean (SD)	RSD
AA (male)	56	4896	35	87.43 (38.79)	44.37%
GD (male)	44	3520	18	80.00 (19.39)	24.24%
KK (female)	44	3927	16	89.25 (25.90)	29.02%
MA (female)	44	5349	18	121.57 (34.16)	28.10%
NE (female)	44	4994	17	113.50 (31.37)	27.64%
PV (male)	44	4659	19	105.89 (37.67)	35.58%
PE (female)	43	2649	19	61.60 (30.22)	49.05%
SV (male)	44	3929	29	89.30 (34.30)	38.40%
SMr (male)	45	1781	17	75.95 (27.42)	36.11%
SMl (male)	43	3266	16	39.58 (34.14)	86.26%
VY (female)	44	3746	18	85.14 (28.11)	33.01%
All set	495	42716	43	86.29 (38.26)	44.33%

$p < 0.001$), and MEntropy (r_{492} = -0.125, p = 0.006) were also significant, but somehow weaker.

Further, we constructed regression model for N_Elements with the 4 factors, which was found to be highly significant ($F_{4,432} = 26.0$, $p < 0.001$), although had rather mediocre $R^2 = 0.194$. Its Akaike Information Criterion (AIC) value was equal to 3100.

$$N\_Elements = 70.5 + 24.9 \times JPEG\_filesize - 12.0 \times PNG\_filesize + 0.253 \times VA\_Elements - 5.4 \times M\_Entropy \tag{1}$$

Since in some cases the visual analyzer failed to produce the metrics, we tested if the model could be constructed without the VA_Elements factor. The regression was found to be highly significant too ($F_{3,488} = 36.1$, $p < 0.001$), although it had somehow lower $R^2 = 0.182$ and poorer AIC = 3498.

$$N\_Elements = 87.4 + 38.5 \times JPEG\_filesize - 24.1 \times PNG\_filesize - 6.2 \times M\_Entropy \tag{2}$$

We used the model (2) to obtain the predicted numbers of elements for the 55 screenshots that the visual analyzer failed to process. Pearson correlation between the predicted values and the actual number of labeled elements (N_Elements) was found to be highly significant, $r_{55} = 0.427$, p = 0.001.

4 Conclusion

Image-based analysis of user interfaces has recognized advantages as it allows taking into account layouts, whitespace, graphical content, etc. independent of the concrete platform and framework used. The visual analysis (HCI vision) software tools generally require lots of training data, particularly for detecting the type of elements in today's manifold web UIs. Relying on internet-based crowdworkers who perform UI labeling is a popular approach for collecting such training data, but controlling the output quality currently involves considerable work overhead. In our work, we proposed to assess data completeness, which in UI labeling equates the number of identified UI elements, via predicting this expected number with metrics automatically calculated for the input image.

For that, we constructed two regression models: (1) relies on the number of UI elements assessed by our dedicated visual analysis tool based on edge detection, while (2) only uses JPEG, PNG and entropy metrics as the factors. The quality of (1) was somehow better, as its $R^2 = 0.194$ was 6.59% higher, for the 1.13 times smaller sample. However, with (2) one is capable of predicting the expected number of UI elements in UI image without the need to rely on external tools.

We see another contribution of the work in the set of pre-defined classes that we devised for web UI labeling and which covered 93.2% of all labeled UI elements in our study. The classes are presented and described in Table 1, and can be used by researchers working on similar problems.

Undoubtedly, the main limitation of our study is lack of the models' testing in real crowdworking to identify workers who undermine completeness in UI labeling tasks. Our future research prospects include collecting the training data for the enhanced visual analyzer through a crowdwork platform and using the models together with outlier analysis to identify neglecting performers.

Another limitation is the relatively low R^2 coefficients in the models, even though (2) allowed to predict the values that had reasonably strong correlation of $r = 0.427$ with the actual number of labeled UI elements. We plan to work on refining the set of factors, probably drawing from the metrics of visual complexity, which is currently extensively studied in HCI.

Our further research prospects also include assessing other dimensions of training data quality, particularly its accuracy, also based on the characteristics of input and output datasets, without the need for extra work effort. For that end, we plan to study the distribution of UI elements' classes and produce the characteristics of the trusted dataset, so that each worker's output could be related to them.

Acknowledgements

The reported study was funded by Russian Ministry of Education and Science, according to the research project No. 2.2327.2017/4.6.

References

1. Semuels, A.: The Internet Is Enabling a New Kind of Poorly Paid Hell. *The Atlantic. Next Economy*, 23 Jan 2018. Accessed 20 May 2019 at https://www.theatlantic.com/business/archive/2018/01/amazon-mechanical-turk/551192/

2. Chen, A.: Inmates in Finland are training AI as part of prison labor. *The Verge*, Mar 28, 2019. Accessed 20 May 2019 at https://www.theverge.com/2019/3/28/18285572/prison-labor-finland-artificial-intelligence-data-tagging-vainu.

3. Daniel, F. et al.: Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1), article 7 (2018).

4. Liu, X. et al.: CDAS: a crowdsourcing data analytics system. *In Proc. of the VLDB Endowment*, 5(10), pp. 1040-1051 (2012).

5. Inel, O. et al.: Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. *In Proc. International Semantic Web Conference*, pp. 486-504 (2014).

6. Fan, J. et al.: iCrowd: An adaptive crowdsourcing framework. *In ACM SIGMOD International Conference on Management of Data*, pp. 1015-1030 (2015).

7. Zheng, Y. et al.: QASCA: quality-aware task assignment system for crowdsourcing applications. *In ACM SIGMOD International Conference on Management of Data*, pp. 1031-1046 (2015).

8. Boychuk, E., Bakaev, M.: Entropy and Compression Based Analysis of Web User Interfaces. *Lecture Notes in Computer Science (International Conference on Web Engineering)*, 11496, pp. 253-261 (2019).

9. Bakaev M. et al.: Auto-extraction and integration of metrics for web user interfaces. *Journal of Web Engineering*, 17(6&7), 561-590 (2018).