The Pennsylvania State University

The Graduate School

INFORMATION EXTRACTION AND RETRIEVAL FROM DIGITAL SCREENSHOTS – ARCHIVING *IN SITU* MEDIA BEHAVIOR

A Thesis in

Information Sciences and Technology

by

Agnese Chiatti

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2019

The thesis of Agnese Chiatti was reviewed and approved* by the following:

Prasenjit Mitra
Professor in Information Sciences and Technology
Associate Dean for Research
Thesis Advisor

Nilam Ram
Professor, Human Development and Family Studies, and Psychology

Xiang Zhang
Associate Professor of Information Sciences and Technology

Mary Beth Rosson
Professor in Information Sciences and Technology
Associate Dean for Graduate and Undergraduate Studies

*Signatures are on file in the Graduate School

ABSTRACT

A significant proportion of individuals' daily activities is experienced through digital devices. Smartphones, specifically, have become one of the preferred interfaces for content consumption and social interaction. Identifying the content that appears on smartphone screens and the rapid switches in content over time is thus a crucial prerequisite to studying media behavior and the potential impacts of screen content on physical, psychological and social health and well-being.

A need then arises, to effectively extract the content enclosed in digital screenshot and represent it in a machine-readable and efficiently retrievable form. Moreover, screenshot images can depict heterogeneous content and applications, making the a priori definition of adequate taxonomies a cumbersome task, even for humans. Privacy protection of the sensitive data captured on screens means the costs associated with manual annotation are large, as the effort cannot be crowd-sourced. Thus, there is need to examine the utility of unsupervised and semi-supervised methods for classifying digital screenshot. This work introduces the implications of applying clustering on large screenshot sets when only a few labeled data points are available.

We present an end-to-end framework implemented to: (i) extract text from digital screenshots, (ii) index the extracted text through Elasticsearch, (iii) store it a MongoDB collection of JSON documents, with their associated metadata, and (iv) classify the screenshot content through a combination of semi-supervised clustering and Active Learning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This work was born as part of the broader *Screenomics* project (http://screenomics.stanford.edu), initially developed at the Center for Advanced Study of Behavioral Science at Stanford University. I would like to acknowledge the direct funders of the Screenomics project, the Stanford University Cyber Social Initiative, the Knight Foundation, and the Pennsylvania State University Colleges of Information Sciences & Technology and Health & Human Development, for providing all the essential data and computational support for the studies reported in this Thesis. I would also like to thank the Screenomics Lab members at large for all the fruitful discussions, either held remotely or during our fun yearly retreats on the West coast. I thoroughly enjoyed being a part of such an inter-disciplinary and diverse team and learned a lot in the process, for which I will always be thankful. Without all of you, this work would have never even seen the light.

This Thesis also happens to be the culmination of an over-2-year long experience as Graduate Student at Penn State University, so a few more *thank you* are in order. I have had so many more masters and mentors that a student could ever wish for. My first thank you is for Dr. Lee Giles for introducing me to PSU and IST and for all his lessons and first-hand anecdotes about the IR world. My deepest thanks also go to Dr. Prasenjit Mitra, for providing his guidance even from afar and for teaching me the single most important lesson: that a researcher's duty is not only to conduct research, but to do so with joy and passion. I was lucky enough to have Dr. Nilam Ram also advising me on this journey. The enthusiasm he can devote not only to the everyday work but also, and more importantly, in building a sense of community and acceptance in the Lab, have always been an endless source of inspiration for me. Thank you, Nilam, for trusting me enough to be a part of the *Screenomics* team, giving me countless opportunities to grow both as an Information Science expert and as human being. I get my new habit of gifting people with chocolate from you. I would also

like to thank Drs. Byron Reeves and Thomas Robinson for giving me roles of technical responsibility in the Screenomics project and tons of food for thought while being exposed to disciplines so far away from mine, and yet so intertwined with it. My most sincere thanks to Dr. Xiang Zhang, for accepting to be in my Committee and carefully revising this Thesis. I would also like to thank Dr. Jian Wu for being a constant reference in the Lab, for patiently putting up with my lack of IT and CS background and for never missing my research presentations. Thanks to Dr. Sagnik Ray Choudhury, for teaching me how to build a search engine from the ground up. Thank you, Scott Pezanowski, for pioneering solutions that looked quite opaque to me in the beginning and then for helping me out in doing the same. Without Kaitlin Oleva Nelson and Jenna Sieber, I would have never been able to wrap up the University paperwork while living abroad; to both go my most sincere thanks. Another huge thanks to my colleagues Xiao Yang and Mimi Brinberg, for proving that we could be a formidable trio on both the East and West coasts, no matter if coding, setting up a happy hour full of delicious cheese within minutes, or getting ready together in near-real time while sharing common spaces.

IST was also a very special and unique place to be in. I would like to thank all the students of the different Cohorts I have overlapped with (with a special mention for the 2016 one!!) for all the corridor chats, GIST gatherings and interesting research discussions in and out of class.

My time at Penn State was spent surrounded by amazing friends, my family away from home. Thanks to Rich, Sam, Prasanna and Sharmila for always being so lovely and the most legendary and royal President and Vice-president couples! Another big thank you is for Stephanie and Zach for choosing me to be a part of their wedding and, above all, for not getting too annoyed when I would get the best Card Against Humanity picks out of blind luck. Thanks to Ning, Nasim, Tiffany, Srishti and all the other brilliant women in IST for the amazing time spent together (and for the memories in San Francisco). Thanks to Mukund and Neisarg for all the laughter and fun (and mango lassi). Thanks to the wise Pratik for being the ultimate gym and all-round life motivator.

Thank you, Shaurya, for being the best dancing partner, for cheering me up even in the darkest times with your faith in me and great sense of humor and for throwing the best parties in 3 square meters covered in led lights humanity has likely ever seen.

Thank you, Chits and Shanjida (and our “adopted” Tracy!), for being the best roommates ever and for all our pizza, movie and Just Dance nights; you have made me feel at home even in the littlest things. To my dearest “vicina” and friend Ana goes a huge thank you; you are marvelous through and through and, without a doubt, the most fun and energetic Maths teacher (and chef!) I am proud to call my friend. Thank you, Kirsten and Travis, for our infamous random encounters on the M bus and for always being such a model of kindness and compassion. I will always be infinitely grateful for my other little box of wonders, Si, for there is nothing she cannot achieve, including lifting the world (and, incidentally, a hundred carton boxes!) off the shoulders of a friend in need.

The biggest thank you is for my little soul sister Saranya for crossing my path and changing it forever. We skipped the small talk in a second and opened a never-ending exchange of whole-hearted conversations, rants, flavored macchiatos, “Ok, but have you eaten yet?”, tears of joy and sadness, arguments, ruinous river falls. It is in our differences that we found where we were the most alike (“Touch wood”). Thank you for always rooting for me and for being my safe harbor. You have showed me how to carry through hardships without indulging in self-pity and, most importantly, that we should better refrain from engaging in outdoor sports together!

The last but most important thank you is for my parents and my brother, for always supporting me, through thick and thin, by enduring through the hardest choices and sacrifices and by celebrating even the smallest successes together. It takes a lot of freedom to be able to express ourselves and find our own path outside the easiest road, and you had the courage to teach me what that freedom is all about, even when it takes the shape of a whole wide ocean lying in between.

Per aspera ad astra.

Chapter 1

Introduction and motivation

Daily experiences are increasingly experienced through digital devices and smartphones, in particular, have become a predominant site for consuming, sharing and searching for media content. Thus, a number of studies of human behavior and "just-in-time" intervention planning can stem from intensive and longitudinal collection of smartphone screenshots, providing evidence of individuals' second-by-second interactions with a wide variety of content.

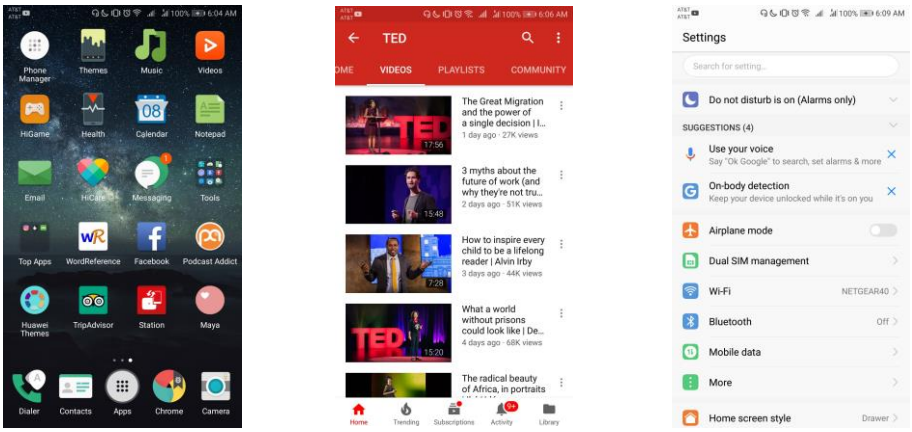
To successfully represent the fast-paced interplay of activities and contents constituting these life threads - with switches occurring as quickly as every 19 seconds [1]- behavioral researchers need a taxonomy of screen content categories. Paradigmatic use cases include, but are not limited to: early disease detection and prevention planning; HCI models of task switching and its implications for attention and memory; ethnographic assessments on the effects of marketing strategies and profiling on low-income populations; studies of political attitudes and voting expressions across social media and news media, fake news detection, and so forth.

A first pre-requisite is then to represent the information enclosed in each captured frame in a way that is machine-readable, efficient to be stored and later retrieved for further analyses. Given the density of textual information screenshots carry along, effective text extraction (i.e., through Optical Character Recognition, or OCR) is a crucial factor contributing to this aim. Furthermore, screenshots provide a unique combination of graphic and scene text, motivating the evaluation of state-of-the-art OCR methods on this particular data set. Given the unique nature of this data set, a need arises for adopting a general-purpose approach, to handle a variety of fonts and layouts, disambiguate icons and graphical contents from purely textual segments, and recognize textual parts which are embedded in advertisements, logos, or video frames. While organizing these

miscellaneous information fragments in a more coherent and interactive repository for further Knowledge Discovery, it is important to consider the organization and accessibility of the extracted data. We particularly focus on the retrieval of screenshot images based on their textual content and metadata, by describing the search engine architecture developed to this aim.

The implemented architecture, and related repository of digital screenshots, are conceived for aiding further analyses on the nature and interplay of contents consumed by media users. As a first effort in that direction, we interrogate on *whether the extracted textual information can be leveraged with visual cues present in each digital screenshot, to classify their main content*. Classification, however, can be cumbersome especially for screenshots (visual snapshots of screens), that include nested data streams (images, text), presented over diverse templates. For instance, if Figures 1-1a and 1-1b are compared, one can see how the proportion of text over icons and graphical content is different. Specifically, in Figure 1-1b, more complex frames (i.e., video previews) are alternated with textual contents and titles. As such, smartphone screenshots form a unique, yet rather unexplored, data typology, providing opportunities to test state-of-the-art methods for data clustering and classification on an unusual collection.

Pattern recognition from a heterogeneous media archive [2], showcasing different topics and digital affordances competing for users' attention [3], with limited prior knowledge and annotation budget available, fosters the search for higher-level and more scalable screenshot representations, through unsupervised and semi-supervised feature learning. Moreover, the confidential nature of screenshots prompts searching for alternatives to crowd-sourcing, when categorizing incoming data streams. Moreover, screenshots enclose visual cues and text in different proportions, based not only on the consumed content (e.g., watching a YouTube video as opposed to reading a news article) but also on the time frame being observed, within the same activity (e.g., as scrolling through a blog post).



(a) (b) (c)

Figure **1-1**: Sample screenshots which would be labeled as: (a) Home screen (b) Youtube, and (c) Settings respectively.

Much evidence has been already provided on the benefits of incorporating multi-modal features when disambiguating and categorizing new data sets [4, 5, 6], however, this hypothesis has not been tested in the specific case of digital screenshots, where a higher degree of varying conditions and entropy is expected, both in visual appearance and textual contents. Consider, for instance, two frames depicting a smartphone home page (Figure 1-1a) and settings menu (Figure 1-1c) respectively. While the two-color palettes are certainly easier to discriminate, even by the human eye, the similar text densities (in terms of number of characters per page) and content overlap (i.e., both frames include the keyword "settings") could have the two mapped to closer feature spaces. Moreover, (i) OCR errors [7, 8] can be propagated to the analysis of the autonomously-extracted text; and (ii) the presence of spurious characters extracted from icons [7] can introduce additional noise to the pipeline, compromising the usability of the produced library [9]. However, one could also argue that the latter characters, even though not carrying over textual meaning *per se*, could aid the discrimination of templates embedding different icon sets. Therefore,

the influence of multi-modal representations on screenshot classification still needs a careful assessment.

In this scenario, defining adequate taxonomies capturing the full spectrum of relevant activities and applications presents additional challenges. Screen behavior is far-reaching, stretching beyond single, domain-specific and task-oriented taxonomies. For instance, generally annotating all sites for social networking as "Social Media" can potentially create a rather miscellaneous set, especially when different media platforms (and interchanging information threads) are concurrently observed, leading to low inter-field agreement on the chosen labels. While experts in media trend analysis might be more interested in the social connotations of content sharing, researchers studying the expression of voting attitudes through social signals might label that same content as "News" or "Events", rather than "Social Media" [10].

A more suitable alternative, is indicating the specific application being used (e.g., Facebook, Instagram, Twitter) or the type of action being performed, when the specific application cannot be inferred (e.g., when watching videos in full screen), explaining the adoption of application-level tags in this work (as shown in Figure 1-1). In fact, this design rationale aids agnostic pattern recognition, i.e., carried out without superimposing field-specific knowledge, ensuring that the learned features are data-driven. Besides, relying on lower-level annotations does not hinder future possibilities to add as many layers of semantic abstraction on top of the chosen taxonomy as required by the specific use case.

In what follows, we present an end-to-end framework to: (i) extract text from digital screenshots, (ii) index the extracted text through Elasticsearch, (iii) store it a MongoDB collection of JSON documents, with their associated metadata, and (iv) classify the screenshot content through a combination of semi-supervised clustering and Active Learning. For the text extraction and image classification components, we also present the specific data preparation and experimental setup used to evaluate the solution.

Chapter 2

Related work

In this Section, the body of literature and relevant related work reviewed is presented and grouped with respect to the specific architectural component being observed. Specifically, we identified these macro-areas of contribution to be: (i) Text Extraction from Imagery, (ii) Content-based Image Retrieval (CBIR), (iii) Visual Pattern Recognition, (iv) Textual Pattern Recognition, (v) Multi-modal Learning, and (vi) Active Learning.

Text Extraction from Imagery

Text in imagery is typically characterized as either machine-printed *graphic* or *scene* text, i.e. captured on objects and native scenes [11]. The overarching goal of Text Information Extraction [11], in both cases, is to first assess the presence of any textual contents, to localize them, and to ultimately recognize the string counterpart of each provided glyph.

Related work has tackled the Text Information Extraction task through a plethora of methods, ranging from morphological operations such as Connected Component Analysis (CCA) [12] and Histogram of Oriented Gradient (HOG) feature extraction [13], to supervised and unsupervised Machine Learning methods like Convolutional Neural Networks (CNN) (standalone [14] or combined with Recurrent Neural Networks to produce more refined models [15]) and Conditional Random Fields (CRF) [16], as well as other classification and probabilistic methods such as Support Vector Machine (SVM) [17] and Markov models [18]. Furthermore, recent studies include the exploitation of Maximally Stable Extremal Regions (MSER) Feature Extraction to identify and localize textual content in imagery [19]. Localization and recognition tasks are

typically integrated, producing an "end-to-end" Information Extraction pipeline [20, 21, 11]. Alternatively, stepwise methodologies [11] detect and classify the candidate textual regions upfront, producing segments of interest to feed the recognition model [22, 14, 23]. While the latter family of methodologies filters out part of the contents and greatly improves the computational efficiency of the process, in principle it can introduce the risk of error accumulation throughout the involved steps, with a more challenging optimization of the single parameters. Conversely, the former integrated architectures can skip the segmentation step, guaranteeing a higher robustness to background complexity, while significantly increasing the computational costs [11].

Even though Text Detection and Optical Character Recognition (OCR) have reached optimal performance on scanned documents (with recognition rates exceeding 99%), the processing of more complex or degraded images is still gathering research interest [11], particularly in the context of natural scene images, ancient manuscripts, and handwritten pieces [20, 15, 13], where accuracy tends to drop around 60%.

In the case of screenshots, some traditional issues are mitigated, e.g., diverse text orientation and uneven illumination, due to the high occurrence of "graphic text" over "scene text". However, other challenges apply, such as co-occurrence of icons and text and the variability in fonts and layouts [11]. Moreover, screenshots represent a hybrid case study, mixing graphic and scene test in varying proportions over time, hence motivating the evaluation of existing techniques on a novel collection.

Content-based Image Retrieval

The extracted text and the associated metadata constitute the basis to represent and retrieve information within the discussed corpus. Image Retrieval can generally be based on the visual elements embedded in the image (i.e., Content-based Image Retrieval or CBIR) or on its textual

metadata, e.g. tags, surrounding captions or headings. This latter branch of Information Retrieval is also known as Concept-based Image Retrieval and, when built over categorical labels, traditionally requires significant manual annotation effort, as opposed to the increased computation complexity implied by CBIR.

Multimedia content recognition and indexing is being already applied: for biomedical imagery cataloging and to assist future diagnosis based on past clinical data [24, 25]; for word matching over scanned and printed document images [26]; for similarity search over large video collections [27].

The work presented in [28] shares some similarities with our current pipeline, however Yeh et al. focus on GUI sub-elements of screenshots, from a user interaction standpoint and without evaluating the OCR robustness in the case of screenshots. Recent work [29] has focused on extracting news articles metadata from smartphone screenshots and exploit those as search parameters, to return the document full text for users to consume at a later stage. To our knowledge, none of the surveyed applications has exploited a combination of text extraction and indexing to retrieve smartphone screenshots of heterogeneous source, offering a diverse range of textual contents (e.g. social media posts, news, articles, health-related threads, text messages, video captions and so forth), and, thus, enabling countless free-text search combinations, over the archived media behaviors.

Visual Pattern Recognition

The need to discover data-driven class taxonomies, the availability of large collections of unlabeled screenshots and the costs associated with manual tagging, are all factors favoring the choice of unsupervised and semi-supervised learning methods for the digital archive at hand.

Unsupervised learning has been traditionally applied with the goal of providing seminal insights on newly-explored image sets, based on the discovered features and feature descriptors. Among fully-unsupervised methods, clustering techniques have been already proven useful, when applied to the categorization of unlabeled image collections, e.g., in [30, 31, 32, 33]. Fully-unsupervised, K-Means clustering has been successfully applied to classify medical images [33] remotely-sensed images [32] or, more broadly, for pixel-wise segmentation [30] and adaptive text recognition [34] in natural scenes. As indicated in the literature, initializing K-Means to randomly-picked centroids can compromise the quality of results. Therefore, more refined extensions of the original algorithm, like K-Means++, are preferable [35]. Clustering methods have been applied independently [30] or enhanced through backward representation Deep learning [31]. Besides Deep Neural architectures, requiring significant computational resources, other recent methods introducing semi-supervision in clustering have experimented on: developing an ensemble of different unsupervised pipelines [36]; incorporating pseudo-labeling or label propagation in the process [37]; estimating the class probability of all unlabeled points to fine-tune feature vectors [38]. Inspired by the aforementioned attempts of combining advantages from both the supervised and the unsupervised realms, in this paper we experimented two pipelines combining K-Means++ clustering with Extreme Gradient Boosting (XGBoost) and Support Vector Machines (SVM) respectively, to control for the application of two different methods when generating probability vectors at each iteration. Both pipelines were conceived to test alternative solutions for class propagation, compared to related methods in sparse subspace clustering [38].

Differently from all the reviewed image collections, screenshots enclose heterogeneous contents whose nature is hard to classify. Nonetheless, we have found it to be a rather unexplored domain. The work conducted in [39], is the most similar found to ours. The approach proposed by Sampat and Haskell comprises a 3-layered convolutional network classifying laptop screenshots over 14 classes. The architecture was pre-trained on the Places205CNN dataset from MIT, to then

transfer the weights over to classifying laptop screenshots. The authors eventually concluded on the inadequacy of the proposed architecture for this classification task. Their approach differs significantly from ours, as the lack of labeled examples was addressed by applying transfer learning in a supervised setting.

Further, dealing with smartphone screenshots, we tackled a relatively different data set from the one used in [39], limiting the analysis to the observation of one activity per frame. In [39], instead, one candidate main window was detected from each screenshot before proceeding with classification, introducing additional marginal entropy in the analysis. Besides, it has been shown that the range of activities performed on smartphones (as diverse as, e.g.: watching videos or reading news in the spare time, but also answering emails on-the-go), and more significantly, the distribution and length of usage sessions [40], are essentially different than on laptops (e.g., typically used for work-related tasks and for more). All these reasons make the digital archive of smartphone screenshots explored in this paper one of a kind.

Textual Pattern Recognition

Distributed representation of words [41] has been widely adopted for textual corpora vector transformations, both in supervised and in unsupervised scenarios.

Recent work [42] has illustrated how combining word2vec representations with more traditional heuristics, i.e., namely TF-IDF, can aid linear separability in classification. Moreover, GloVe [43], a semantic vector space model based on term co-occurrence counts, has been shown to outperform the word2vec probabilistic approach on specific NLP sub-tasks, e.g. word analogy. When dealing with images, pattern recognition is typically performed with the image itself as exclusive input modality, without including text-derived features in the representation. We conjecture this trend to be mainly associated to: the lower incidence of textual portions (compared

to pictorial elements) in traditional image collections (i.e., consisting of natural images), and the performance implications of applying OCR and pattern recognition on cascade. In fact, text extraction is error-prone [9, 8], and, in the particular case of screenshots [2], can lead to (i) duplicated characters caused by overlapping segmented areas, (ii) spurious glyphs extracted from visual elements (e.g., GUI icons), and other recognition errors mainly ascribable to (iii) color, contrast and morphological similarity between background and foreground elements, (iv) presence of "text in the wild", i.e., natural scene text.

On the contrary, OCR-extracted text has been effectively exploited for topic modeling and knowledge discovery on libraries of scanned documents [44], i.e., where textual portions contribute significantly to the overall data structure and contrast and lighting conditions simplify text extraction [11]. Nonetheless, even in the case of purely textual documents, the successful extraction of graphical elements (mainly charts and diagrams) has enhanced semantic interpretation of the available data collections [45], further complicating the formulation of hypotheses on the relative impact of visuals and text to result quality. Additionally, several data-dependent considerations usually apply.

In the specific case of screenshots, the question of whether to include text-derived features in the pattern analysis process becomes even more debatable, given their hybrid nature compared to the two other opposite ends of the spectrum: scanned documents and natural scenes [2,46]. In the context of Screen Content Image (SCI) quality assessment, it has been shown how the distribution of pixel naturalness in screen content images [46] is statistically different from both its natural scene and purely-textual counterparts. However, to the best of our knowledge, no prior work has clarified the relative impact of text-derived features to the validity measurement of screenshot clusters, nor OCR-extracted, unstructured text has been included in the image feature vector.

Multi-modal Learning

Joint-learning of image-text embeddings [4] has been applied to image datasets associated with transcribed captions to improve entity localization. Recent work has shown how multi-modality can lead to better knowledge capture, especially when extrapolating low-level concepts from the represented entities [47]. In this context, text was pre-processed by applying known techniques in Text Classification, e.g. word2vec [41] or GloVe [43], to be more uniformly merged with the image vectors [48]. In other cases [49], visual patterns and textual features have also been jointly learned to improve automated event extraction.

However, to our knowledge, none of the reviewed multi-modal approaches have been tested on digital screenshots. In the related literature, these methods are typically applied on image caption learning [50], question answering tasks [48], or sematic triple representation learning [51]. Instead, in the proposed approach we include the full textual contents as a feature and measure the relative impact of multi-modal feature sources on the resulting cluster validity.

Active Learning

Active Learning methods have been thoroughly explored, to compensate for the lack of human annotation budget or, more broadly, maximize a task-oriented utility function. While many methods simply concentrated on single-valued definitions of model uncertainty, often combined with committee-based disagreement assessments, margin-based methods have provided a more refined alternative for image classification tasks [52]. Classic definitions of margin with respect to SVM decision boundaries have also been adapted to the clustering case, to select the minimum margin for points that are most equidistant from their two closest subspaces [53]. Further, the introduction of batch methods exploiting dynamic programming algorithms has facilitated scale

towards larger data sets. To our knowledge, all of the surveyed methods have yet to be applied to digital screenshot images. Inspired by different contributions, summarized in [54] and [53], and in the attempt to apply the lessons learned on high-resolution remote sensed images, we customized an open-source implementation for Active Learning, that combines batch selection of informative points with diversity requirements, to suit the framework presented in this work.

Chapter 3

Digital Screenshot collection

The reference data set for the experiments presented in the following sections is the result of an ongoing study of media behavior, which, at the time of said experiments, had already collected data on device use across 101 participants located in the US, collectively providing for 5'532'739 smartphone screenshots. Full-resolution screenshots were taken every 5 seconds, bundled and encrypted for secure transmission to a Cloud-based centralized database.

Images are organized in buckets (i.e., by subject ID) and associated with the related timestamps of capture. Screenshots are further processed and page layouts segmented to feed an Optical Character Recognition (OCR) module. Ultimately, one JSON document is produced for each input screenshot to store the extracted textual contents, when any textual content is found (see also Chapter 4 for further details).

Data collection was conducted with full IRB for Human Subjects Research approval and participants gave explicit consent for monitoring. Privacy protection is certainly a primary concern while handling this collection: (i) data transmission from and to the Cloud-based server were encrypted, (ii) only authorized personnel trained in human subject research handling of sensitive data were given access to the data, and (iii) specific data handling rules were enforced, also for the human annotators, who labeled data within a secure server, to avoid downloading images to their local machines.

Chapter 4

Text Extraction from Digital Screenshots

Implemented workflow

To enhance the quality of extracted text, we setup a procedure to process the raw screenshots and feed the OCR engine with graphic segments where the textual content could be more easily distinguished from the background. This data preparation routine was built on top of the OpenCV library for Image Processing [55]. The overall workflow is depicted in Figure 4-1.

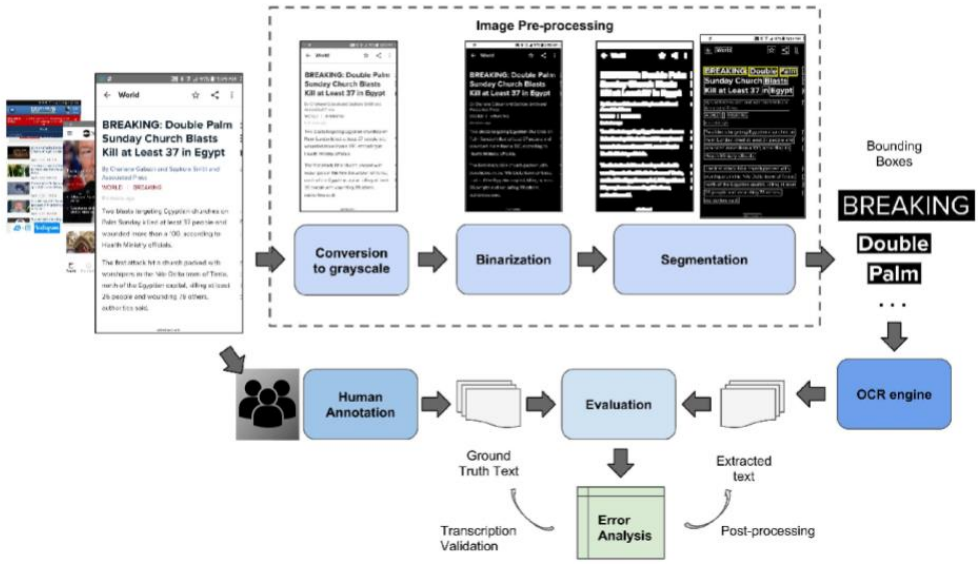


Figure **4-1**: Workflow applied to extract text from digital screenshots. Input images are (i) first converted to grayscale, (ii) and binarized, (iii) to then recognize individual bounding boxes wrapping the candidate text regions and (iv) feed them to the OCR module. In parallel, ground truth transcriptions were manually-collected for a subset of screenshots.

Conversion to grayscale

Conversion of the images from RGB to grayscale is a prerequisite to binarization, which ultimately leads to better discrimination of the foreground from the background (i.e., the end goal of text/object detection).

Binarization

Binarization methods are conceived to transform grayscale images to binary format (i.e., black and white). The association of each pixel to black or white pixel sets can typically follow a global or a local approach to adaptive thresholding [56]. In the former case, the cutoff is constant within the same image, whereas in the latter case the threshold value can vary based on the local illumination conditions of the image. Since we could assume a uniform illumination relative to each screenshot, we applied a combination of simple inverse thresholding with Otsu's global binarization. Finally, we skipped the skew estimation step, given the predominantly horizontal layout of the target text, thus leading to a more scalable pre-processing of the incoming images.

Segmentation

This step identified rectangular bounding boxes wrapping the textual content of the images, i.e., the standard shape format used by existing OCR engines for text detection [57].

We adopted a Connected Component based approach (similar to the methodology followed in [12]): (i) the white pixels were first dilated to create more organic white regions (Figure 4-1), (ii) the uniform regions were then detected, and (iii) a rectangle was drawn around the identified area.

To limit the duplicated recognition of the same regions (i.e., when a smaller rectangle is completely enclosed in a larger area), we included an additional check to filter out the innermost

rectangles. However, overlapping rectangles were still identified (Figure4-3d), leading to partially duplicated text, when bounding boxes were ultimately passed to the OCR engine.

Optical Character Recognition (OCR)

After pre-processing, each segmented region was fed to the OCR engine, using the Python wrapper for Tesseract. Tesseract recognizes text in a "two-pass process" [21] that integrates character segmentation with the recognition module and uses backtracking to improve the quality of the output. First, attempts are made to recognize single words separately. Second, part of the words (based on a quality evaluation) are then passed to an adaptive classifier as training data. This increases the ability of the classifier to recognize the remainder of the text in the page.

We relied on the stable release of Tesseract (v. 3.03) for our first OCR run, but the alpha release of Tesseract 4.0 was also tested on the considered sample set. The timing of our analysis provided an opportunity to compare the baseline engine with an updated approach, which integrates a LSTM-based module for line recognition in the pipeline, in a similar fashion to the OCRopus framework [23]. We were then able to assess the improvement introduced by a Neural-Net component, without increasing the computation and time expenses. Tesseract 4 has already been trained on approximately 400,000 lines of text that include 4,500 fonts. We exploited the pre-trained OCR architecture as-is, without re-training.

For each screenshot, we produce a JSON document, including, besides the extracted text in full: the subject ID, a unique identifier for the screenshot (i.e., combining subject ID and the timestamp of capture), other information extracted on the image through off-the-shelf tools, namely number of faces identified (if any), image entropy and probabilities associated with the presence of specific App logos, as thoroughly described in [58].

Workflow evaluation

For evaluation purposes, we selected a subset of screenshots for study, testing, and refinement of text extraction procedures. Specifically, we partitioned a random subsample of 17 participants and applied reservoir sampling to randomly select 150 images from each day, thus accommodating the fact that the size of each daily collection of screenshots is not known a priori. The analysis set here consists of 13,172 smartphone screenshots representative of a typical user's behavior.

Gold standard text was needed to ultimately evaluate the quality of the text extracted from the screenshots. Gold standard data is often collected through crowdsourced transcription services (e.g., Amazon Mechanical Turk), or, in some cases, through third-party commercial APIs for text extraction (e.g., Microsoft Cogntive Services). However, as our data are highly sensitive and require privacy protection, we customized a free, and open-source localturk tool [59] and involved three human annotators trained in human subjects' research and confidentiality.

The human annotators securely accessed a dedicated Virtual Machine hosting the transcription tool. Figure 4-2 showcases the GUI interface used for transcribing the images. The GUI's left side showed the full screenshot, annotated with a set of bounding boxes produced by our Image Pre-processing module, while the individual boxes requiring transcription were displayed on the right side.

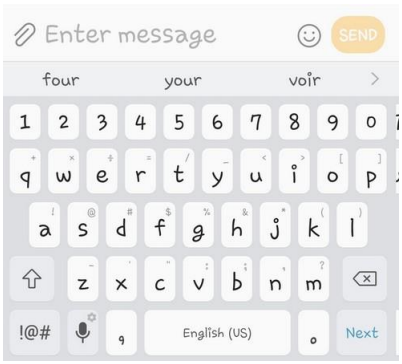
Figure **4-2**: GUI for the customized tagging tool (illustrative screenshots).

Bounding boxes were presented to annotators in the same scanning order followed by our Image Pre-processing module when detecting the rectangular segments to be passed to the OCR engine, i.e., from top to bottom and from left to right. This precaution ensured consistency between the collected ground truth text and the generated text to be evaluated.

The annotators followed detailed instructions to complete the transcription tasks. Specifically, annotators were instructed to preserve capitalization and punctuation, use special character marks to disambiguate icons and graphic contents from text, and notate locations where text spanned beyond the segmentation cuts. The complete image with bounding boxes allowed annotators to check and note if any portions of text had been missed by the bounding box procedure and were not included in the transcription request. Similarly, partially overlapping boxes could arise in the set (Figure 4-3d). Annotators were instructed to take the full picture as reference, to transcribe the overlapping text only once, and to mark any cases that were not covered by the instructions. These

checks (i.e., ordering of bounding boxes, handling of overlapping bounding boxes, updating transcription rules dynamically, unanticipated situations) facilitated the subsequent analysis of segmentation inaccuracies. In this first experimental setup, each annotator transcribed an independent set of images. In other words, agreement across different annotated sets was not evaluated. However, after a first run of evaluation and error analysis, the ground truth collection was cross-checked and manually corrected. This validation step supported the later assessment of the degree of human-error embedded in the process.

Figure **4-3**: Examples of discovered patterns after error analysis (illustrative screenshots): (a) fancy fonts, (b) text embedded in videos is extracted more effectively when integrating the pre-processing routine, (c) smartphone upper banners add marginal noise, (d) inaccurate segmentation can lead to overlapping bounding boxes.



(a)

(b)

(c)

(d)

Leveraging quality with the expensiveness of the transcription task [60], we obtained complete transcriptions of 1,360 screenshots. In sum, comprehensive ground truth data to evaluate our text extraction pipeline was produced for 10 percent of the analysis sample.

Evaluation Metrics

To evaluate the quality of our generated text against the gold standard transcribed text, we defined OCR accuracy both at the character and at the word level, as a complement of the error rates, and further discriminating between order-dependent and order-independent error rate at the word-level evaluation.

Word Error Rate (WER)

WER is based on the Levehnstein distance [61] between the provided text and the reference text:

$$WER = \frac{i_w + s_w + d_w}{n_w}$$

Where i_w, s_w and d_w refer to the number of words to be inserted, substituted and deleted to transform the given text into the reference text. Among all the possible permutations, the values that minimize the sum $i_w + s_w + d_w$ are considered. The resulting number of transformations is then normalized by the number of words in the reference text (n_w).

Character Error rate (CER)

Similarly, the same metric can be defined at the single character level:

$$CER = \frac{i + s + d}{n}$$

where word counts are replaced by character counts, following the same rationale defined for the WER metric. Thus, the error rate will typically be higher at the word level, as failures in recognizing single characters ultimately impact the recognition of complete words.

Position-independent WER (PER)

The position-independent WER is based on a bag-of-words approach that does not take the provided order of words into account when evaluating the word-level error rate. Hence, this metric loosens one of the constraints of the standard WER. As a result, this metric tends to overestimate the actual word-level accuracy.

Hence, improving the accuracy of our text extraction framework essentially translates into minimizing the error rates, both at the character and at the word level. Furthermore, we prioritize the improvement on the standard WER over its positional-independent counterpart, with the intent to preserve the syntactic and semantic attributes of the extracted text and support meaningful Natural Language Processing and Topic Evolution analyses of the obtained documents.

To record these three metrics, we used the ocrevalUAtion open source tool [62]. Besides the described metrics, this tool also reports error rates by character, and aligned bitext for each document match, facilitating the comparison of the generated output against the reference text [62]. We set up the evaluation so that punctuation and case differences were counted as well, when calculating the scores. Finally, computational time was recorded and evaluated as an additional factor contributing to the overall process efficiency. We compared the performance of two different releases of the Tesseract OCR engine before and after validating the manually transcribed text. This experimental setup aimed to quantify the improvement introduced by a NeuralNet-based module for line recognition (i.e., the additional component introduced with Tesseract 4 compared to Tesseract 3.03) when the other parameters were kept constant.

Evaluation Results

First, we compared the two Tesseract-based solutions, applied as-is, with the end-to-end framework of Figure 4-1. As illustrated in Tables 4-1 and 4-2, the Image Pre-processing framework proposed here improved the overall accuracy, both at the character and word level [7]. Hence, this improvement justifies the additional computational steps implied by the proposed solution.

Table **4-1**: Comparison of baseline Tesseract 3 and 4: before and after applying the noise removal heuristic.

Approach	**Character-level**		**Word-level**			
	ER	Accuracy	ER	Accuracy	PER	Accuracy
Baseline Tesseract 3	36.13%	63.87%	47.21%	52.79%	38.78%	61.22%
Baseline Tesseract 3 + heuristic	33.45%	66.55%	43.42%	56.58%	36.01%	63.99%
Baseline Tesseract 4	**33.68%**	**66.32%**	**41.54%**	**58.46%**	**33.93%**	**66.07%**
Baseline Tesseract 4 + heuristic	**31.73%**	**68.27%**	**38.38%**	**61.62%**	**31.46%**	**68.54%**

.

Table **4-2**: Comparison of Tesseract 3 and 4 with Image Pre-processing: before and after applying the noise removal heuristic.

Approach	**Character-level**		**Word-level**			
	ER	Accuracy	ER	Accuracy	PER	Accuracy
ImgProc+Tesseract 3	33.02%	66.98%	44.50%	55.50%	39.63%	60.37%
ImgProc+Tesseract 3 + heuristic	31.95%	68.05%	41.26%	58.74%	37.33%	62.67%
ImgProc+Tesseract 4	**31.15%**	**68.85%**	**40.27%**	**59.73%**	**35.56%**	**64.44%**
ImgProc+Tesseract 4 + heuristic	**30.68%**	**69.32%**	**38.95%**	**61.05%**	**35.06%**	**64.94%**

Table **4-3**: Comparison of Tesseract 3 and 4 with Image Pre-processing, after correcting the human-annotated scripts.

Approach	**Character-level**		**Word-level**			
	ER	Accuracy	ER	Accuracy	PER	Accuracy
ImgProc+Tesseract 3	27.42%	72.58%	39.12%	60.88%	33.81%	66.19%
ImgProc+Tesseract 3 + heuristic	27.35%	72.65%	37.67%	62.33%	32.09%	67.91%
ImgProc+Tesseract 4	**25.16%**	**74.84%**	**33.85%**	**66.15%**	**28.77%**	**71.23%**
ImgProc+Tesseract 4 + heuristic	**25.69%**	**74.31%**	**34.38%**	**65.62%**	**28.76%**	**71.24%**

After a careful inspection and error analysis of the outputs produced in the two cases, we identified a lower robustness of the pre-processing framework in the presence of peculiar fonts chosen as a default on users' phones (an example of such cases is shown in Figure 4-3a), as opposed to the baseline alternatives (i.e., in the absence of screenshot pre-processing). This observation suggested that the binarization and segmentation parameters can be further fine-tuned to improve the handling and recognition of font sets that are used by specific users. On the other hand, the proposed framework enhanced the recognition of text that is embedded in video frames (as depicted in Figure 4-3b), when compared to the baseline performance of the two Tesseract releases.

Further, we wanted to discriminate between the accuracy deficiencies caused by the human error inherent to the transcription process and the error rates directly associated with the adopted Image Processing framework. Table 4-2 shows the results obtained when applying Tesseract 3.03 and Tesseract 4 to the pre-processed smartphone screenshots, i.e., after segmenting the regions of interest that were candidates for carrying fragments of textual content. The introduction of the LSTM-based line-recognition component slightly improved the OCR accuracy both at the single-character and word level.

Analysis of the returned errors showed that the most prominent faults seem associated with: (i) the presence of *icons* and *other graphic features* in line with the text, (ii) defects in the *reference transcriptions*, (iii) presence of *peculiar fonts*, (iv) textual contents that are *difficult to distinguish from their backgrounds* (e.g., both are a light color), and (v) *partially overlapping segmented regions* leading to duplicated characters. Examples of all scenarios are illustrated in Figure 4-3.

To quantify the incidence of the first category of errors, we developed a naive heuristic to post-process our text and filter out the first line when it matches specific regular expressions that plausibly represent the top banner of smartphone displays. An example of the critical regions and text generated by the OCR engine when text was mixed with icons is provided by Figure 4-3c.

Specifically, top lines were removed only when other textual content besides the upper banner was found. This process elicited a better measure of net accuracy on the textual contents of interest, by eliminating marginal and noisy content. However, when the top banner (i.e., typically icons, clock and battery level) was the only textual information included in the screenshot, it was not filtered out. Overall, applying this heuristic provided a more reliable proxy of the actual accuracy obtained by the current framework.

Inherent human error associated with our manual transcription procedures also contributed to quality loss. Thus, the produced transcriptions were validated manually and the evaluation step repeated after correction. The results, depicted in Table 4-3, demonstrate the significant incidence of inadequate reference text on the overall scores, when compared to Table 4-2. Typical transcription faults include the occurrence of typos or oversights leading to incomplete transcriptions of the actual text. Due to these partial transcriptions, text, which was correctly recognized by the OCR engine, was absent from the reference text, artificially increasing the error rate. These inaccuracies were corrected through a posterior validation check. Please note, part of the error and burden related to transcriptions, was caused by the transcription tool's GUI and procedural instructions, which need further refinement, based on the observations collected during this exploratory phase.

After removing the transcription error effect from the set, the solution which integrates an LSTM-based line-recognition system still provided the highest performance. Tests with Tesseract 3.03 and 4 were run in parallel on two identical Debian, quad-core Virtual Machines. The computation times to process the sample (i.e., 13,172 phone screenshots), without applying *ad hoc* training in either of the two cases, were comparable for all pipelines. In sum, there were not any notable tradeoffs between Tesseract 3.03 and 4.0 in terms of process efficiency.

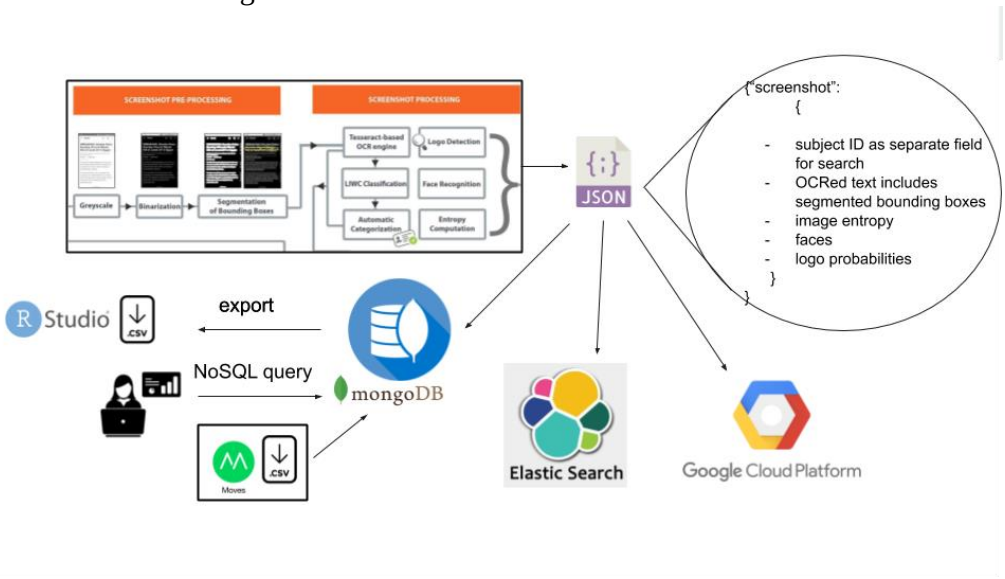
Chapter 5

Content-based Screenshot Retrieval

Data organization

The output of the information extraction module supplements the raw screenshots with a collection of additional "metadata", as described in Chapter 4. Furthermore, we kept track of the original image file locations and stored it as part of the created JSON document as well, so that the retrieved images, all stored in on a Google Cloud server bucket, can be visualized if necessary. Heterogeneity of data types is accommodated using a secure, limited access NoSQL database deployed in accordance with human subject protocols for protection of research participants' privacy. Text, image, numeric, string, spatial, and temporal data are fused (often by subject and time) within a schema-less NoSQL framework for flexible query and analysis. We use the open source MongoDB document-oriented framework that facilitates expansion of the metadata associated with subsets of the collection as different researchers in our group develop, refine, and add new fields and corresponding metrics to the feature set. This data management framework (also illustrated in Figure 5-1) is specifically constructed to facilitate scaling, including repository expansion, parallelization, flexible workload distribution, and smooth integration with search, retrieval, and data analysis technologies.

Figure **5-1**: Framework followed to organize the extracted information and link it with the original image collection (maintained on secured Google Cloud Platform servers). Each screenshot is represented as a JSON document (refer to Chapter 4) first, and then stored in a MongoDB database, conveniently synchronized with RStudio to allow the analysts and involved researchers to flexibly export ad hoc data reports or run their analyses directly on the document collection. When available, subject data are also enhanced with location information (input as csv files generated by the Moves app). Ultimately, the JSON format is natively suited to be indexed through ElasticSearch, which constitutes the backend of our specialized screenshot search engine.



Specialized Search Engine

Examination of the document store is facilitated by a custom search engine that allows a user to enter a textual query (e.g., “president AND New York Times”) that returns a ranked list of screenshot thumbnails related to the input query. Indexing and search is done using a tailored vertical search engine built using ElasticSearch and Apache Solr Lucene. In brief, the JSON document associated with each screenshot is indexed with respect to its enclosed text (with stemming and ignoring stop words) and content fields (e.g., geohash, content categories). Solr Lucene exploits an inverse indexing approach. As opposed to a forward index, an inverted index consists of keyword-centric entries, referencing the document containing each considered term.

When a user enters a query into the web-based user interface, all images containing text matching the input query are drawn from the document store, ranked based on relevance and displayed to the user as a list of relevant screenshots. In the current implementation we used the Okapi B25 metric [63], to rank document by relevance. For a document *D* and query *Q* containing q_i,..., q_n keywords, the document-query similarity score is then defined as follows:

$$\text{score}(D,Q) = \sum_{i=1}^{n} idf(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avdl}\right)}$$

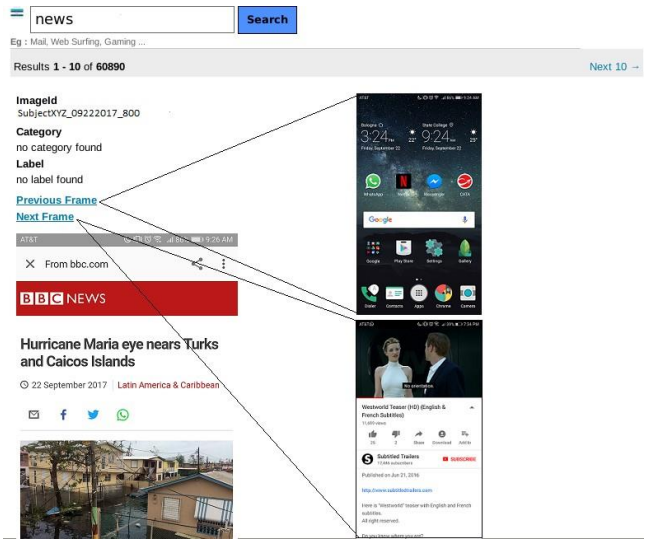
where *avdl* is the average document length and k_l and *b* are free parameters (kept to default values in this case). The latter Equation is based on the same underlying principles of the term frequency-inverse document frequency *tf-idf* heuristic, when determining the relevancy of words within each considered document. Particularly, the *idf* score recipient for each keyword q_i is computed as:

$$idf(q_i) = \log\frac{N-n(q_i)+0.5}{n(q_i)+0.5}$$

where $n(q_i)$ represents the number of documents containing q_i and *N* is the total size of the corpus. *Tf-idf* increases proportionally with the frequency of occurrence of a term within a document and is normalized based on the term occurrence throughout the whole corpus. However, BM25 has been proven more effective than *tf-idf* in mirroring the user's perceived relevancy towards returned contents, due to the introduced regularization parameters [63].

The obtained ranked results are ultimately presented to the user through a dedicated Web interface, implemented in Python Django (Figure 5-2), only accessible by personnel trained on Human Subject Research and currently running ethnographic studies on the observed subjects.

Figure **5-2**: Search Engine Graphic User Interface (illustrative screenshots).



For each retrieved image, certain metadata are listed, including the id, times- tamp and category. Each thumbnail can be expanded to full-size resolution, aiding further exploration of the retrieved textual content. Summaries and links accompanying each search hit provide additional information (e.g., content category, geographic location, links to temporally adjacent screenshots). The search engine is critical for understanding the range of screen behaviors that pertain to specific content areas (e.g., health, politics) and generating hypotheses about how the content embedded in each screenshot may relate to a wide range of thoughts, actions, and feelings.

The baseline architecture followed to for the implemented search engine has been publicly released and is available at https://github.com/achiatti/search-engine-template.

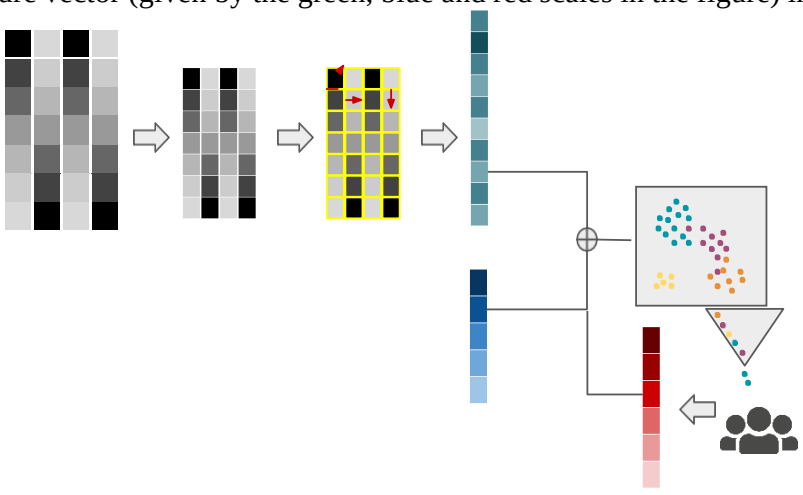
Chapter 6

Multi-modal Screenshot Classification through Active Learning

Implemented Workflow

The proposed workflow is illustrated in Figure 6-1 and combines K-Means semi-supervised clustering with informative and diverse batch selection from a pool of unlabeled samples. The label information gathered by querying a human oracle is then leveraged with weaker probability estimates produced by a supervised classifier, for all the remaining points in the unlabeled set.

Figure **6-1**: Proposed framework for representing and clustering smartphone screenshots. Moving clockwise from lower left, screenshot images are vectorized and dimensionally reduced using gray-scale conversion, histogram of gradients, and principal components analysis. In parallel, as shown in center of figure, the textual content embedded in the screenshots (previously extracted using OCR) is vectorized using GloVe. Then, as shown on the right side of the figure the visual feature vector (green scale) and text vector (blue scale) are concatenated and clustered through K-means, combined with informative and diverse active learning to query a human oracle. The manually-annotated labels are then fed to a supervised classifier (either XGBoost or SVM) to obtain a class probability vector (red scale) for all the unlabeled data points. From the second iteration onward, class probabilities are added to the feature vector (given by the green, blue and red scales in the figure) input to clustering.



K-Means Clustering

Historically introduced for signal processing, K-means is a partitional clustering method attempting to identify as many group centroids as specified by users through a dedicated parameter (K). Different setups can be chosen for a more robust centroid initialization, e.g. K-Means++ [35]. Points are then clustered as a result of being assigned to their closest centroid. Centroids are updated based on the formed clusters, with assignments and updates repeated until convergence of the cluster configuration. The proposed framework applies unsupervised K-Means to select an initial batch of points through Active Learning. In subsequent iterations, labeled and unlabeled features are integrated into the vector representations so that clustering proceeds in a semi-supervised manner.

Informative and Diverse Active Learning

Active Learning methods can aid the selection of points to be labeled, and thus minimize the annotation effort. Two criteria measuring the optimality of selected batches with respect to some utility function (or labeling cost), are informativeness and diversity. While informativeness quantifies the ability to reduce the uncertainty on the underlying clustering model, another important contributing factor is ensuring disparity across the selected instances to mitigate potential redundancies. The experimented framework adapts a released routine in AL[1] to integrate it with K-Means clustering, to take both factors into account. First, informativeness is measured in a margin-based fashion.

[1] https://github.com/google/active-learning

Intuitively, lowest margin points imply a higher statistical uncertainty. Thus, the N margin values are sorted in ascending order and selected greedily up to exhausting a user-determined batch size n. Further, data points are added to the batch only if their cluster distribution mirrors the overall cluster distribution. This dynamic batch selection step accommodates for the second requirement, i.e., preserving diversity over the underlying data distribution.

Class Probability Propagation

To integrate the ground truth knowledge produced on each Active Learning iteration, the oracle-annotated examples are used to train a supervised classifier, which will estimate probabilities for the remaining unlabeled points. The generated probability vectors are then concatenated with the original vector representation, to update K-Means centroids accordingly. Further, higher weights are associated to the probability vectors, than to the other features composing the input, to purposefully boost the incidence of probabilities predicted from the oracle-annotated examples.

To control for the computational cost and required training set size at each iteration (i.e., favoring shallow approaches as opposed to Deep supervised architectures), and to test for model adaptation from a limited set of labeled points, we embedded an ensemble-based method, namely Extreme Gradient Boosting (XGBoost) in one of the pipelines under evaluation, and compared it against a baseline routine which relied on Support Vector Machines (SVM).

XGBoost

Extreme Gradient Boosting [64] is a supervised model that generates predictions from input n-dimensional vectors, based on the observed classes. It summarizes judgments from an ensemble of classification and regression trees (CART) that carry decision values on the leaf nodes.

Operationally, the predictions are summed across different trees, in the attempt to increase the accuracy of the classifier. Thus, while the mathematical modeling of XGBoost is mostly comparable to Random Forest models, the actual training routine implemented in the two cases is different. In fact, the so-called tree boosting routine optimizes performance (defined through a specific objective function) at each iteration, to refine the training of each subsequent decision tree.

Support-vector Machine (SVM)

Given input n-dimensional vectors, an SVM classifier [65] estimates a hyperplane that can best group the data points into distinct classes. As a result, linear SVM does not ensure optimal separation for non-linear data. Typically, one adopted countermeasure is kernel-based learning, i.e., mapping the input vector features into a different mathematical space, to improve their separability [66]. In the discussed framework, we integrated one commonly-used kernel setting based on the Radial Basis Function (RBF). This transformation function is derived from computing the radial Euclidean distance between points and centroids and normalizes the input vectors.

Workflow evaluation

Data preparation

From the overall set, we selected a randomized sample, stratified across all subjects, by applying reservoir sampling with batch size 500. We chose reservoir sampling as randomization strategy (i) because each subject provided for a different number of screenshots and (ii) to optimize selection across a significantly large collection. As a result, either 500 images were picked from each bucket (i.e., subject ID), when the considered set exceeded 500 units, or the whole set was

taken otherwise. Once the randomized screenshot sample was identified, the related JSON documents enclosing auxiliary metadata and textual contents (extracted through the top-performing routine identified in [7]) were retrieved as well.

The selected screenshot sample is representative of different subjects, during consistent use captured at different time points. Therefore, we expected that even images that would be unanimously associated to the same cluster would enclose some degree of background and content heterogeneity. Thus, to normalize the input image distribution and reduce sensitivity to marginal noise, we first represented screenshots as 1280x720 grayscale-pixel vectors, resized the vectors to 256x256 resolution and, finally, performed value standardization to obtain a normally-distributed input.

Visual Feature Description

To represent screenshots through a set of distinctive visual components, we adopted a state-of-the-art method for SIFT Object Recognition, i.e., Histogram of Gradients (HOG) [67] as Visual Feature Descriptor. A grid of 8x8 pixel cells was built on top of the 256x256 images, to compute pixel-wise gradients and group them together based on 9 orientation bins. The sliding window size was set to blocks of 2x2 cells, to obtain a flattened representation of each screenshot as a one-dimensional vector consisting of 34'596 distinct features.

Dimensionality Reduction

Vector representations obtain\-ed from the previous step were then reduced to a more compact form, to optimize computation across a large sample set. We initially applied Principal Component Analysis (PCA) over the first 1000 components to assess the optimal number of

eigenvalues for the subsequent runs. To optimize the computational cost and leverage memory constraints emerging from PCA application to a high-dimensional matrix, we chose the randomized Singular-value Decomposition (SVD) option based on the work by Halko et al. [68]. We converged towards a certain number of dimensions after inspecting the cumulative sum of variances explained by the top-n components. As a result, the selected threshold indicates the cutoff after which including additional components does not lead to increased cumulative variance. Specifically, we found the number of discovered principal components to be equal to 225 in the standalone image vectors case and to 897 in the joint image-text representation scenario.

Text processing

For each input screenshot, the textual content extracted through a combination of Image Processing and OCR [7] is also retrieved, when available (i.e., for all screenshots that contained any text in the first place). First, we applied whitespace-based tokenization to the raw text and removed punctuation. Then, all the resulting tokens were reduced to lowercase. }

Text Vector Representation

To obtain a vectorized text representation of comparable density to the aforementioned visual feature vectors, we utilized GloVe pre-trained 300-dimensional word embeddings derived from the Wikipedia 2014 and Gigaword 5 data sets and accounting for 6 Billion tokens in total [43]. Word embeddings were combined through the following workflow: (i) for each token in the screenshot text, the corresponding semantic vector was retrieved, if the considered token was found in the reference vocabulary, (ii) a zero-valued vector was associated to each unknown token, (iii) the overall vector describing the full document was obtained from the weighted average of word

embeddings identified in the first phase, using TF-IDF values as weights, as frequency-based metrics were recently found to improve linear separability across word embeddings [42]. Finally, (iv) zero-valued vectors were associated to all the screenshots that did not contain any text.

Parameter Sensitivity Analysis

To converge towards a specific number of clusters without super-imposing a prior class cardinality (a required parameter when initializing K-Means or other partitional clustering variants) we run a sensitivity analysis for the K parameter with respect to the Sum of Squared Errors (SSE) curve.

For algorithms using Euclidean distance, accuracy can be derived from the sum of the squared errors (SSE), based, in this case, on the distance between data items and their centroids. Specifically, the SSE values are averaged across all clusters in the configuration.

We applied the elbow method for K selection on the SSE curve, after drawing the curve on a 0-1000 range in iterations of 10 units, as illustrated in Figure. In brief, we selected the value of K corresponding to the 80% break of the SSE curve, as the marginal gain after the identified point would not justify setting a higher number of clusters. In our case, the cutoff value was 190 clusters.

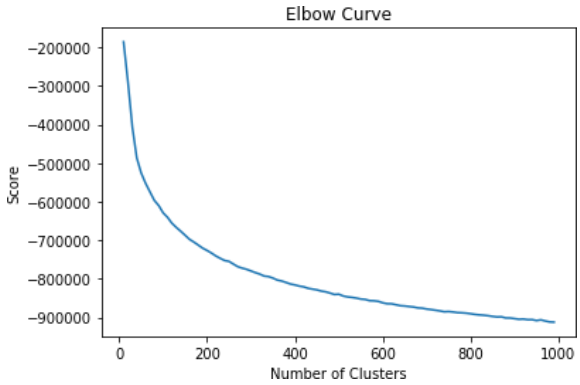


Figure **6-2**: SSE values (averaged across all clusters in the configuration) for 0 < *K* 1000 over iterations of 10 units. The number of clusters (K) to initialize K-Means is selected based on 80% break of the blue curve, i.e., corresponding to 190 clusters in this case.

Implementation details

In the discussed experiments, we exploited K-Means++ for centroid initialization and set K to 190, as explained in the previous Section. In all the experimented setups, similarity of points is represented through Euclidean distances. The Informative and Diverse Active Learning batch size was set to 200 over 10 iterations, leading to 2000 labels produced by the oracle, for each experimental pipeline. Three different pipelines were run based on different combinations of input vector features and supervised classifiers used to estimate class probabilities for the unlabeled points on iterations. Specifically, (i) XGBoost and (ii) SVM classification were applied on the vectorized images, and, ultimately, (iii) the first workflow was tested on joint image-text vectors as well, to assess the performance effects of introducing multi-modal features.

Screenshots selected by the Active Learning module were presented to two Graduate Students in Informatics and Industrial and Manufacturing Engineering with prior knowledge of the data characteristics and both attaining to a given taxonomy, consisting of 60 classes. Labels were formulated at the app-level of granularity (e.g., Facebook, Twitter, Reddit), or, when the content could be associated with any smartphone app, based on the type of action being represented (e.g., Navigation, Web search, Notifications), as introduced in Section 1. Exemplary screenshots and their related category are showcased in Figure 1-1.

Evaluation metrics

Given the semi-supervised setting, internal cluster validity metrics were used to assess the impact of both the input representation (i.e., not including text vectors/including text vectors) and the embedded classifier (i.e., XGBoost, SVM) for class probability propagation. The quality of the

output configuration, across all three selected indices, is formulated by jointly measuring intra-cluster cohesion and inter-cluster separation.

Silhouette Index

Introduced in [69], the silhouette index is used for graphical evaluation of the within-cluster cohesion, when leveraged with adequate separation across different clusters. For each data point i in the input set, let *a(i)* be the average distance between i and all the remainder points lying in the same cluster. Further, one can derive *b(i)* as the minimum average distance between i and all points in any other cluster. The Silhouette value for i is then defined as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

Thus, averaging *s(i)* across all data points provides estimates about the overall quality of the clustered configuration.

Dunn Index

Another validity index is the Dunn Index [70]. For each cluster of n-dimensional vectors C_i, where *k =1,...,K*, Δ_i indicates the maximum inter-vector distance:

$$\Delta_k = \max_{x,y \in C_k} d(x, y)$$

Further, let $\delta(\mathrm{C_I}, \mathrm{C_j})$ be the minimum distance between clusters C_i and C_j (formulated here as the distance between the two closest data points in the two clusters):

$$\delta(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

The Dunn-Index mathematical formulation for K clusters can be then derived as:

$$DI_K = \frac{\min_{1 \le i < j \le K} \delta(C_i, C_j)}{\max_{1 \le k \le K} \Delta_k}$$

Davies-Bouldin Index

Similarly to the Silhouette and Dunn indices, the Davies-Bouldin Index [71] is another internal clustering evaluation metric. In the Euclidean distance scenario, let C_i be a cluster of n-dimensional vectors where *i=1,...,N*, S_i measures the scatter within each cluster the Euclidean distance between vectors and their centroid. Further, for two clusters C_i and C_j, let $M_{i,j}$ indicate the Euclidean distance between the two cluster centers as proxy of separation between clusters C_i and C_j. The Davies-Bouldin Index for K clusters is then derived, as a function of the ratio between the two said components:

$$DB_K = \frac{1}{K} \sum_{i=1}^{K} \max \frac{S_i + S_j}{M_{i,j}}$$

By definition, the Silhouette and Dunn indices are maximized to improve the end cluster quality, whereas, conversely, local minima are to be identified in the Davis-Bouldin curve.

Evaluation Results

First, we applied the framework to the image vectorized representation (i.e., without including the text representation). Figure 6-3 shows the cluster validity scores obtained when

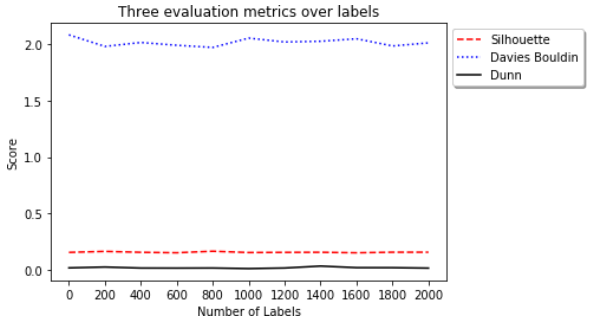
integrating an SVM classifier with RBF kernel, over 10 total iterations and for incremental batches of 200 labels. After an initial marginal improvement from the starting scores, cluster validity keeps fluctuating in the same range of the score obtained with the first unsupervised configuration, i.e., before any labeled examples are introduced. This observation holds across all three validity metrics, although marginal variations in cluster validity from one iteration to the next is more significant in the Davies-Bouldin case. When feeding the same image vectors to the XGBoost-based pipeline instead, we obtained comparable results, with respect to the range of scores obtained, across all evaluation metrics (Figure 6-3b).

Nonetheless, it can be noticed, comparing the shape of curves in Figure 6-3b with the ones in Figure 6-3a, that the XGBoost-based alternative was more robust to variations in the propagated class probabilities, from one iteration to the next, signaling that exploiting tree ensembles made the learning less dependent on the number of labeled points, i.e., on the judgements espressed by the oracle.

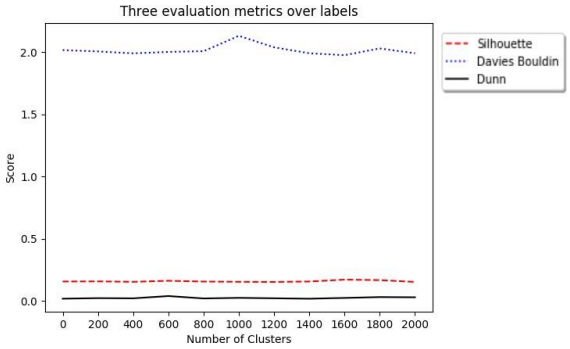
However, when the same pipeline embedding XGBoost was tested on joint image-text embeddings (Figure 6-3c), the validity scores where again more subject to fluctuations from iteration to iteration, i.e., as can be seen by comparing the shape of all three curves in Figure 6-3c with the curves in Figure 6-3b. More significantly, adding text embeddings to the feature set led to a performance decrease with respect to all three evaluation metrics. In fact, the Davies-Bouldin scores (marked in blue), which should be minimized, increased from Figure 6-3b to Figure 6-3c.

Conversely, the Silhouette (in red) and Dunn scores (in black), which ideally should be maximized, decreased from Figure 6-3b to Figure 6-3c.

(a) Visual feature vectors with SVM class probability propagation.



(b) Visual feature vectors with XGBoost class probability propagation.



(c) Multi-modal feature vectors with XGBoost class probability propagation.

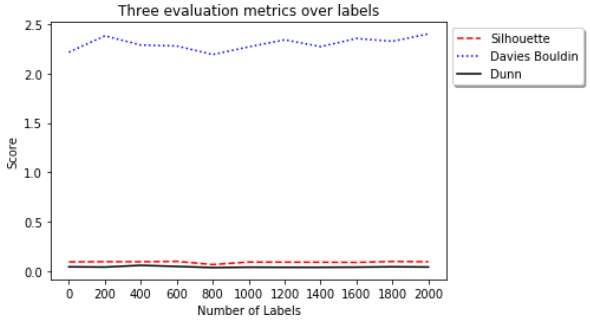


Figure **6-3**: Cluster validity results, evaluated with respect to the Silhouette (red curve), Davies-Bouldin (blue curve) and Dunn indices (black curve), over the number of labeled data points. Optimal results correspond to lower scores in the blue curve and higher values for the red and black curves respectively. The range (i.e., from minimum to maximum) of scores, which is comparable in Figures 4a and 4b, differs in Figure, indicating that the introduction of text-derived features hindered the overall performance. On average, these results hold for all considered indices.

These results indicate that, although potentially helpful to disambiguate images enclosing the same set of application icons over varying backgrounds (e.g., in the "Home screen" case, Figure 1-1a), introducing OCR-derived textual features negatively impacted the clustering performance. Thus, with respect to the two opposite hypotheses outlined in the introductory sections, when discussing the marginal utility of OCR-extracted text for disambiguating different classes, the experiments ultimately highlighted a more significant incidence of the negative contributing factors (i.e., errors embedded in the OCR-extracted text [2, 8] and shared terms across different contents, as noticeable, for instance, from Figures 1-1a and 1-1c).

Furthermore, one could argue that the presence of spurious glyphs in the raw input text was not enough to improve the separation of different icon sets. However, these results could also be ascribed to the specificity of the pre-trained GloVe corpora, when contrasted with the peculiar terms which are specific to smartphone screens. All cases, nonetheless, showed a quite persistent performance even after increasing the size of the labeled sample.

Chapter 7

Conclusion

In this work, we firstly introduced a complete workflow for text extraction and retrieval from smartphone screenshots. The pipeline is based on OpenCV image-processing and Tesseract OCR modules. We evaluated the quality of the extracted text and showed how word and character accuracy improved through refinement of image pre-processing procedures and NeuralNet based line-recognition system introduced in the newly released Tesseract 4.0. Detailed analysis of word and character errors suggest that further improvements are possible, both generally and in the data production process. Additional error analyses identified and isolated the most prominent factors contributing to quality loss. Ultimately, a search engine application was developed based on the inherent characteristics of the data at hand, for the immediate use for the involved analysts.

Human error embedded in the ground truth data production process was present and unanticipated. The findings that correction of human errors provided improvements in accuracy that were of similar size as other technical refinements suggests some reconsideration of how ground truth (and training) data are produced for new data streams. Given the costs, in terms of time and accuracy, there is much incentive to develop iterative solutions that reduce human involvement in the loop to correction of automatically-generated transcriptions.

Future work on the more technical aspects of the text extraction process for screenshots include fine-tuning and sensitivity analysis of the Image Pre-processing parameters, which should increase the solution's robustness in the presence of diverse fonts, embedded icons, and mixed graphic contents.

Preprocessing may be particularly important for text extraction from laptop screens, for which the method is generalized, due to the possibility that multiple windows may be visible in a

laptop screenshot. Problematically, each window might have different background and foreground contrasts, thus significantly introducing marginal noise. Thus, methods that effectively and accurately partition and segment the main active window will need to be developed and refined.

As well, the results of this study suggest there is some need for re-training Tesseract-based solutions so that they better handle characters and words that only appear in specific sub-portions of the data (e.g., individual users' device preferences and idiosyncratic font use). These additions and accommodations will expand computation complexity and time and therefore need to be evaluated with respect to added value.

We have also presented exploratory analyses on clustering smartphone screenshots based on multi-modal features, i.e., the visual features of the examined images and the textual features extracted in the prior phase.

Methodologically, we applied techniques that have been applied successfully to process other classes of high-resolution images, e.g., medical [33] and remotely-sensed frames [32], both in unsupervised settings, and combined with active learning when label availability is limited [53]. In our experiments, however, semi-supervised clustering was applied in combination with two alternative supervised classifiers for class probability propagation, thus following an alternative approach to sparse subspace-based semi-supervised clustering [38].

We found that performance, expressed in terms of cluster validity, was comparable with respect to the employed classifier and can be further ameliorated. Further, the current representation of textual features, on average, led to a slight performance decrease, confirming multi-modal learning incorporating textual embeddings to be highly data dependent and, hence, further distinguishing digital screenshot from other data collections treated in the literature.

These evidences can serve as baseline for future improvement, particularly about: (i) revising the strategies and methods used to create *ad hoc* image-text embeddings, (ii) enriching the taxonomy used for annotating screenshots, after conducting error analyses on the obtained clusters,

and (iii) extending the image annotation protocols. In the experimented setting, labels were simply provided by human annotators, based on the given taxonomy and on their sole judgment. However, refinements in how human oracles are asked to formulate judgments would mitigate their individual biases. One solution would be to present annotators with a set of top candidate label suggestions to choose from, to automatically pre-filter the available options.

The seminal experiments presented in this paper have confirmed the unique nature of smartphone screenshot collections. Although, the intuition is that more data/features are better for classification, in the presenting setting this is not yet supported by theoretical and foundational proofs, in the related literature. Here, as is also the case in some other low signal-to-noise scenarios, additional information added more noise than useful information. Including a second modality of features generated overburden and ultimately lead to lower-quality results in classifying smartphone screenshots. Although not fully conclusive, the results are novel specifically because of the counter-intuitive finding, and the indication that the screenshot setting is different than other state-of-the-art questions about recovery of image captions [48]. Generally, the move into social-science relevant data will require additional reconsideration of the standard intuitions. This work begins that effort and opens opportunity to test alternative frameworks for clustering digital screenshots from limited annotated examples.

In sum, here we have presented the end-to-end architecture used to build a comprehensive archive of digital screenshots, which can be further queried and analyzed to inquiry on media behavior as it occurs on its ubiquitous platforms, i.e., *in situ*. Screenshots are a new kind of document that appears to hold high value for studying human behavior, as pointed out in the overview journal paper representing the "manifesto" of *Screenomics* [58]. As the pioneer exploration into extracting useful information from the first screenshot repository, the application of Image Pre-processing, OCR and CBIR tools was successful. While we are only at the beginning of learning what these data hold, these initial results are promising and produce excitement about

how Information Extraction methods may be adapted for these data and contribute to new knowledge about how, when, and why individuals engage with digital life.

REFERENCES

[1] Leo Yeykelis, James Cummings, and Byron Reeves. 2014. Multitasking on a single device: Arousal and the frequency, anticipation, and prediction of switching between media content on a computer. *Journal of Communication* 64 (2014), 167–192. Issue 1.

[2] Agnese Chiatti, M. J. Cho, Anupriya Gagneja, Xiao Yang, Mimi Brinberg, Katie Roehrick, Sagnik Ray Choudhury, Nilam Ram, Byron Reeves, and C. Lee Giles. 2018. Text Extraction and Retrieval from Smartphone Screen- shots: Building a Repository for Life in Media. In *Proceedings of the 33rd ACM/SIGAPP Symposium on Applied Computing (SAC 2018)*. ACM.

[3] S Shyam Sundar, Saraswathi Bellur, Jeeyun Oh, Qian Xu, and Haiyan Jia. 2014. User experience of on-screen interaction techniques: An experimen- tal investigation of clicking, sliding, zooming, hovering, dragging, and flipping. *Human–Computer Interaction* 29, 2 (2014), 109–152.

[4] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure- preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5005–5013.

[5] Yagmur Gizem Cinar, Susana Zoghbi, and Marie-Francine Moens. 2015. Inferring user interests on social media from text and images. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE, 1342–1347.

[6] Sergio Oramas, Francesco Barbieri, Oriol Nieto, and Xavier Serra. 2018. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21*. (2018).

[7] Agnese Chiatti, Xiao Yang, Mimi Brinberg, M. J. Cho, Anupriya Gagneja, Nilam Ram, Byron Reeves, and C. Lee Giles. 2017. Text Extraction from Smartphone Screenshots to Archive in situ Media Behavior. In *Proceedings of the 9th International Conference on Knowledge Capture (K-CAP 2017)*. ACM.

[8] Myriam C Traub, Thaer Samar, Jacco van Ossenbruggen, and Lynda Hard- man. 2018. Impact of Crowdsourcing OCR Improvements on Retrievability Bias. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. ACM, 29–36.

[9] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, Muriel Visani, and Jean-Philippe Moreux. 2017. Impact of OCR errors on the use of digital libraries: towards a better access to information. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*. IEEE Press, 249–252.

[10] Omar Alonso, Vasileios Kandylas, and Serge-Eric Tremblay. 2018. How it Happened: Discovering and Archiving the Evolution of a Story Using Social Signals. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. ACM, 193–202.

[11] Qixiang Ye and David Doermann. 2015. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence* 37, 7 (2015), 1480–1500.

[12] Kamrul Hasan Talukder and Tania Mallick. 2014. Connected component based approach for text extraction from color image. In *Computer and Information Technology (ICCIT), 2014 17th International Conference on*. IEEE, 204–209.

[13] Kai Wang and Serge Belongie. 2010. Word spotting in the wild. In *European Conference on Computer Vision*. Springer, 591–604.

[14] Zhu, Yuanping, Jun Sun, and Satoshi Naoi. "Recognizing natural scene characters by convolutional neural network and bimodal image enhancement." International Workshop on Camera-Based Document Analysis and Recognition. Springer, Berlin, Heidelberg, 2011.

[15] Wenyi Huang, Dafang He, Xiao Yang, Zihan Zhou, Daniel Kifer, and C Lee Giles. 2016. Detecting Arbitrary Oriented Text in the Wild with a Visual Attention Model. In *Proceedings of the 2016 ACM on Multimedia Conference (MM '16)*. ACM, 551–555.

[16] Zhang, Hongwei, et al. "An improved scene text extraction method using conditional random field and optical character recognition." 2011 International Conference on Document Analysis and Recognition. IEEE, 2011.

[17] Sheshadri, Karthik, and Santosh Kumar Divvala. "Exemplar Driven Character Recognition in the Wild." BMVC. 2012.

[18] Weinman, Jerod J., et al. "Toward integrated scene text reading." IEEE transactions on pattern analysis and machine intelligence 36.2 (2013): 375-387.

[19] Shi, Cunzhao, et al. "Scene text detection using graph model built upon maximally stable extremal regions." Pattern recognition letters 34.2 (2013): 107-116.

[20] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. 2012. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21ist International Conference on Pattern Recognition (ICPR'12)*. IEEE, 3304–3308.

[21] Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, Vol. 2. IEEE, 629–633.

[22] Elagouni, Khaoula, Christophe Garcia, and Pascale Sébillot. "A comprehensive neural-based approach for text recognition in videos using natural language processing." Proceedings of the 1st ACM International Conference on Multimedia Retrieval. ACM, 2011.

[23] Thomas M Breuel. 2008. The OCRopus open source OCR system. In Electronic Imaging 2008. International Society for Optics and Photonics, 68150F–68150F.

[24] Andreia V Faria, Kenichi Oishi, Shoko Yoshida, Argye Hillis, Michael I Miller, and Susumu Mori. 2015. Content-based image retrieval for brain MRI: An image- searching engine and population-based analysis to utilize past clinical data for future diagnosis. *NeuroImage: Clinical* 7 (2015), 367–376.

[25] Byung K. Jung, Sung Y. Shin, Wei Wang, Hyung D. Choi, and Jeong K. Pack. 2014. Similar MRI Object Retrieval Based on Modified Contour to Centroid Triangulation with Arc Difference Rate. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC '14)*. ACM, New York, NY, USA, 31–32.

[26] Million Meshesha and C. V. Jawahar. 2008. Matching word images for content- based retrieval from printed document images. *International Journal of Document Analysis and Recognition (IJDAR)* 11, 1 (01 Oct 2008), 29–38.

[27] Henrique Batista da Silva, Raquel Pereira de Almeida, Gabriel Barbosa da Fonseca, Carlos Caetano, Dario Vieira, Zenilton K. Gonçalves do Patrocínio, Jr., Arnaldo de Albuquerque Araújo, and Silvio Jamil F. Guimarães. 2016. Video Similarity Search by Using Compact Representations. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC 16)*. ACM, New York, NY, USA, 80–83.

[28] T. Yeh, T. Chang, and R. C. Miller. 2009. Sikuli: using GUI screenshots for search and automation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 183–192.

[29] Kazutoshi Umemoto, Ruihua Song, Jian-Yun Nie, Xing Xie, Katsumi Tanaka, and Yong Rui. 2017. Search by Screenshots for Universal Article Clipping in Mobile Apps. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 34.

[30] Fan Hu, Gui-Song Xia, Zifeng Wang, Xin Huang, Liangpei Zhang, and Hong Sun. 2015. Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8, 5 (2015).

[31] Jianwei Yang, Devi Parikh, and Dhruv Batra. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5147–5156.

[32] Biplab Banerjee, Francesca Bovolo, Avik Bhattacharya, Lorenzo Bruzzone, Subhasis Chaudhuri, and B Krishna Mohan. 2015. A new self-training- based unsupervised satellite image classification technique using cluster ensemble strategy. *IEEE Geoscience and Remote Sensing Letters* 12, 4 (2015), 741–745.

[33] S Julian Savari Antony and S Ravi. 2015. A new approach to determine the classification of mammographic image using K-means clustering algorithm. *International Journal of Advancements in Research & Technology* 4, 2 (2015).

[34] Xu-Cheng Yin, Wei-Yi Pei, Jun Zhang, and Hong-Wei Hao. 2015. Multi- orientation scene text detection with adaptive clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 9 (2015), 1930–1937.

[35] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.

[36] Y. Zhao and M. K. Hryniewicki. 2018. XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[37] Yi, Yugen, et al. "Label propagation based semi-supervised non-negative matrix factorization for feature extraction." Neurocomputing 149 (2015): 1021-1037.

[38] Qing Yan, Yun Ding, Yi Xia, Yanwen Chong, and Chunhou Zheng. 2017. Class-probability propagation of supervised information based on sparse subspace clustering for hyperspectral images. *Remote Sensing* 9, 10 (2017), 1017.

[39] Anand Sampat and Avery Haskell. 2015. CNN for task classification us- ing computer screenshots for integration into dynamic calendar / task management systems. (2015).

[40] Antti Oulasvirta, Tye Rattenbury, Lingyi Ma, and Eeva Raita. 2012. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing* 16, 1 (2012), 105–114.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their composi- tionality. In *Advances in neural information processing systems*. 3111–3119.

[42] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*. IEEE, 136–140.

[43] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[44] Carlos G Figuerola. 2018. Applying topic modeling techniques to degraded texts: Spanish historical press during the Transición (1977-1982). In *Pro- ceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*. ACM, 857–862.

[45] Rabah A Al-Zaidy and C Lee Giles. 2017. A Machine Learning Approach for Semantic Structuring of Scientific Charts in Scholarly Documents. In *AAAI*. 4644–4649.

[46] Huan Yang, Yuming Fang, and Weisi Lin. 2015. Perceptual quality assess- ment of screen content images. *IEEE Transactions on Image Processing* 24, 11 (2015), 4408–4421.

[47] Fabian Both, Steffen Thoma, and Achim Rettinger. 2017. Cross-modal Knowledge Transfer: Improving the Word Embedding of Apple by Looking at Oranges. In *Proceedings of the Knowledge Capture Conference (K-CAP 2017)*. Article 18, 8 pages.

[48] Bolaños, Marc, et al. "VIBIKNet: Visual bidirectional kernelized network for visual question answering." Iberian Conference on Pattern Recognition and Image Analysis. Springer, Cham, 2017.

[49] Tongtao Zhang, Spencer Whitehead, Hanwang Zhang, Hongzhi Li, Joseph Ellis, Lifu Huang, Wei Liu, Heng Ji, and Shih-Fu Chang. 2017. Improving Event Extraction via Multimodal Integration. In *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 270–278.

[50] Hui Mao, Ming Cheung, and James She. 2017. DeepArt: Learning Joint Rep- resentations of Visual Arts. In *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 1183–1191.

[51] Fudong Nian, Bing-Kun Bao, Teng Li, and Changsheng Xu. 2017. Multi- Modal Knowledge Representation Learning via Webly-Supervised Rela- tionships Mining. In *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 411–419.

[52] Devis Tuia, Michele Volpi, Loris Copa, Mikhail Kanevski, and Jordi Munoz- Mari. 2011. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing* 5, 3 (2011), 606–617.

[53] John Lipor and Laura Balzano. 2015. Margin-based active subspace clus- tering. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*. IEEE, 377–380.

[54] Begüm Demir, Claudio Persello, and Lorenzo Bruzzone. 2011. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 49, 3 (2011), 1014–1031.

[55] Itseez. 2015. Open Source Computer Vision Library. https://github.com/itseez/ opencv. (2015).

[56] Oeivind Due Trier and Anil K. Jain. 1995. Goal-directed evaluation of binarization methods. *IEEE transactions on Pattern analysis and Machine Intelligence* 17, 12 (1995), 1191–1201.

[57] Yi Lu. 1995. Machine printed character segmentation: An overview. *Pattern recognition* 28, 1 (1995), 67–80.

[58] Byron Reeves, Nilam Ram, Thomas N. Robinson, James J. Cummings, C. Lee Giles, Jennifer Pan, Agnese Chiatti, Mj Cho, Katie Roehrick, Xiao Yang, Anupriya Gagneja, Miriam Brinberg, Daniel Muise, Yingdan Lu, Mufan Luo, Andrew Fitzgerald & Leo Yeykelis (2019) Screenomics: A Framework to Capture and Analyze Personal Life Experiences and the Ways that Technology Shapes Them. *Human–Computer Interaction*, pp.1-52.

[59] Dan Vanderkam. [n. d.]. localturk. https://github.com/danvk/localturk. ([n. d.]).

[60] Julius Schöning, Patrick Faion, and Gunther Heidemann. 2015. Semi-automatic Ground Truth Annotation in Videos: An Interactive Tool for Polygon-based Object Annotation and Segmentation. In *Proceedings of the 8th International Conference on Knowledge Capture (K-CAP 2015)*. ACM, New York, NY, USA, Article 17, 4 pages. https://doi.org/10.1145/2815833.2816947

[61] V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10 (Feb. 1966), 707.

[62] Rafael C Carrasco. 2014. An open-source OCR evaluation tool. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*. ACM, 179–184.

[63] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[64] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[65] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[66] Bernhard Schölkopf and Alexander J Smola. 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[67] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1. IEEE, 886–893.

[68] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Find- ing structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* 53, 2 (2011), 217–288.

[69] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[70] J. C. Dunn. 1974. Well-Separated Clusters and Optimal Fuzzy Partitions. In *Journal of Cybernetics* 4, 1 (1974), 95–104.

[71] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), 224– 227.