DHQ: Digital Humanities Quarterly
2020
Volume 14 Number 2

Crowdsourcing Image Extraction and Annotation: Software Development and Case Study

Ana Jofre <jofrea_at_sunypoly_dot_edu>, SUNY Polytechnic
Vincent Berardi <berardi_at_chapman_dot_edu>, Chapman University
Kathleen P.J. Brennan <KPJBRENNAN_at_GMAIL_dot_COM>, University of Queensland
Aisha Cornejo
Carl Bennett
John Harlan

Abstract

We describe the development of web-based software that facilitates large-scale, crowdsourced image extraction and annotation within image-heavy corpora that are of interest to the digital humanities. An application of this software is then detailed and evaluated through a case study where it was deployed within Amazon Mechanical Turk to extract and annotate faces from the archives of *Time* magazine. Annotation labels included categories such as age, gender, and race that were subsequently used to train machine learning models. The systemization of our crowdsourced data collection and worker quality verification procedures are detailed within this case study. We outline a data verification methodology that used validation images and required only two annotations per image to produce high-fidelity data that has comparable results to methods using five annotations per image. Finally, we provide instructions for customizing our software to meet the needs for other studies, with the goal of offering this resource to researchers undertaking the analysis of objects within other image-heavy archives.

1. Introduction

The amount of multimedia data available is steadily increasing [James 2014], which has led to many instances where it is desirable to identify and annotate objects located within an image. Examples include the detection of features from outdoor cameras [Hipp et al. 2013] [Hipp et al. 2015] and the classification of animal species [Welinder and Perona 2010] [Caltech-UCSD Birds 200 2018]. Machine learning and other quantitative methodologies can be used to identify objects within images (see [LeCun et al. 2015] for an example), but their complexity and the requirement for an optimized training set often limit the use of these approaches. A viable alternative is crowdsourcing, the process of enlisting untrained individuals to perform computationally intensive tasks, which has been extensively used in a variety of projects [Kuang et al. 2015] [Manovich et al. 2016] [Yu et al. 2013] [Tomnod 2018] [Clickworkers 2018]. Amazon's Mechanical Turk (AMT) service for crowdsourcing work is popular with many researchers across disciplines and allows requesters to post tasks, and matches these tasks with anonymous workers who complete them. 1

Our specific interest is in identifying and labeling images of faces from the *Time* magazine archive to gain historical insight on American cultural trends. Collecting such a data set requires a two-step process: 1) identify all faces within the corpus and 2) annotate each face according to standardized protocols for feature designation. In this paper, we detail the development of a web-based image-cropping and annotation software for performing these tasks, and we describe our rigorous verification methods for both the cropping and the annotation. Notably, we developed a verification procedure that required only two annotations per image to produce high-fidelity data. The data collected using the methods described here was then used to train an object detector and image classifier machine learning models. 2

Our methods are illustrated through a case study where the software was used to crop and label human faces from an archive of *Time* magazine. While our web-based interface is platform-independent, it was administered as an external 3

survey link on AMT. The design process, details of our data collection methods, and instructions for others to customize the software to crop alternative objects from other archives are all described. The software and methodology described here has been used for our own digital humanities project [Jofre et al. 2020a] [Jofre et al. 2020b], and we believe it may be useful to other researchers.

2. Motivation and Background

This work was motivated by an interest in using large, image-heavy corpora, in particular periodical archives, to gain insights into cultural history. Interpreting large cultural corpora requires both quantitative methods drawn from data science and qualitative methods drawn from technology, cultural, and social studies. From this perspective, we are interested in questions concerning what the faces in a magazine archive could reveal about the larger, historical context of a publication, questions such as how gender/race/age representation have changed over time, and how these correlate with the magazine's text and with broader cultural trends. 4

The archive under consideration for our case study consists of approximately 4,500 issues from *Time* magazine, ranging from 1923 through 2014. The corpus comprises approximately 500,000 .jpg files, with each page of each issue, including the cover, representing one file. We selected *Time* magazine for a number of reasons. First, while there are a few existing studies of this corpus (see [de Souza 2014] and [Manovich and Douglass 2009]), there is certainly more work to be done on the visual aspects of the archive by moving beyond the cover images and text. Second, *Time* has been a mainstay publication in the United States for nearly a century, and in that period has witnessed vast cultural, political, and technological changes. Third, it has a relatively well documented corporate history (see [Prendergast and Colvin 1986]), which allows us to examine the internal context of the production of the magazine vis-à-vis its external context like wars, political movements, changes in fashion, and so on. Finally, the *Time* corpus is widely held in library collections in the United States, and available online through *The Vault* at https://time.com/vault/. 5

The data we collected using the crowdsourcing methods described in this paper has been published as a dataset in the Journal of Cultural Analytics [Jofre et al. 2020a], available for use to all researchers in the digital humanities. We used the crowdsourced data to train an algorithm to extract all the images of faces from our *Time* magazine archive and classify their gender. The high-granularity of the automatically generated data allowed us to undertake a detailed study on gender representation in *Time* magazine [Jofre et al. 2020b]. 6

Previous studies have successfully used crowdsourcing to achieve goals similar to ours. For instance, when examining features of traffic intersections, the correlation between crowdsourced results and experts was 0.86 for vehicles, 0.90 for pedestrians, and 0.49 for cyclists [Hipp et al. 2013]. Similarly, when assessing 99 Flickr images for the presence of 53 features, the correlation between crowdsourced results and expert raters was 0.92 [Nowak and Rüger 2010]. In both of these studies, crowdsourced labels were derived by averaging the labels produced by multiple individuals. While these correlations are encouraging, there are known challenges associated with crowdsourcing. Occasionally, crowdsource workers have been shown to arbitrarily select answers or give vague responses in an effort to complete jobs more quickly [Downs et al. 2010]. This behavior can be reduced by adding verifiable qualification questions, often called honeypots, to crowdsourcing procedures [Kittur et al. 2008]. Furthermore, the demographics of crowdsource workers are typically skewed towards low income workers from India and the United States, who tend to be young and female [Casey et al. 2017]. Our data collection crowdsourcing methods are mindful of concerns about the potential of inadvertently exploiting low-visibility and/or vulnerable populations and intentionally aim to provide reasonable compensation (for further discussion of these issues, see [Irani 2015]). 7

While there are many other solutions for researchers seeking to perform image extraction and annotation via crowdsourcing, we believe that our software fills a unique niche for humanities researchers who want to have full control of the data collection and quality controls. Most solutions are geared towards machine learning researchers and provide these services as a bundle, where the client receives the requested clean data. These include LabelBox (https://labelbox.com/product/platform), LionBridge (https://lionbridge.ai/services/image-annotation/), Hive (https://thehive.ai/), Figure Eight (https://www.figure-eight.com/), and Appen (https://appen.com/). Such black-box solutions are not suitable for the humanities, where we must be mindful of who is doing the tagging. Our software allows 8

the researcher to track individual workers to examine their effect on the data. Furthermore, it is platform-independent, allowing it to be deployed on any crowdsourcing site. We are aware of one other standalone image cropping and tagging software package, labelImg (https://github.com/tzutalin/labelImg), but it is not web-based, which limits its deployment.

The software package and methodology we developed are intentionally flexible, both in the corpora they can analyze and in the crowdsourcing platform on which they can be deployed. For the former, our motivation was to allow our tools to be used with a variety of sources, such as the Look Magazine archive, hosted by the Library of Congress [Look Magazine 2012]. For the latter, we did not want to exclusively link the project to AMT because we want the option of using other crowdsourcing platforms. 9

3. Development and Deployment of Interface

3.1 Determination of Image Features to Be Assessed

In preliminary work, project leaders identified the following nine facial features of interest: 1.) Gender, classified as Male, Female or Unknown; 2.) Race, classified according to current U.S. census categories as American Indian, Asian, Black, Pacific Islander, White, or Unknown; 3.) Emotion, classified according to Ekman's six basic emotions as Anger, Disgust, Fear, Happy, Sad, or Surprise (Ekman and Friesen 1986); 4.) Racial Stereotype, classified as Yes or No; 5.) Magazine Context, classified as Advertisement, Cover, or Feature Story; 6.) Image Type, classified as Photograph versus Illustration; 7.) Image Color, classified as Color or Black & White; 8.) Multiple Faces in the Image, classified as Yes or No; and 9.) Image Quality, classified as Good, Fair, or Poor. 10

One issue from each of the ten decades spanned by the data (1920s-2010s) was selected at random and analyzed by student research assistants. The student coders proceeded through all pages in an issue (range: 50-160), identified faces, and annotated the features according to the above categories. Throughout this process, coders were asked to keep track of anomalous faces that were not easily classified, a process that was extremely valuable in refining our procedures. For example, due to the presence of animal faces and masks, the operational definition of a classifiable face was changed to human faces where at least one eye and clear facial features are present. Single color images required the Image Color classification levels to be changed to Color versus Monochrome and an "Author" category was added to Magazine Context. There was little agreement among raters concerning the presence of stereotypes and facial emotions, so these categories were eliminated. The emotion variable was replaced by a binary Smile variable and a Face Angle variable (whether the face is in profile or facing the viewer). Furthermore, most of the Unknown labels for the Race and Gender categories were assigned to babies or young children, so a binary Adult variable was also added. The final list of facial features is provided in Table 1. 11

With the updated feature list established, three coders reviewed a single issue and annotated the 185 faces that were identified by all three individuals when reviewing the issue. To assess interrater reliability (IRR), Cohen's kappa (κ) was calculated for each facial category. κ values between 0.60 and 0.75 are typically interpreted as representing good agreement while κ > 0.75 characterizes excellent agreement. The average κ was 0.809 and all values were above 0.721 with the exception of image quality, with $\kappa = 0.363$. These values are summarized in Table 1. This IRR exercise revealed that, when reviewing the coder data for pages with multiple faces, it was challenging to make interrater comparisons since it was often difficult to determine which exact face corresponded with a given set of labels. This led to a major revision in our protocol where, rather than having individuals annotate faces and store the results as they reviewed pages, they would first crop each face, so that each set of assigned labels could be associated with a specific cropped face. 12

Variable Name	Classification Options	Cohen's κ
Adult	Yes or No	0.771
Face Angle	Profile or Straight	0.819
Gender	Female, Male, or Unknown	0.932
Image Color	Color or Monochrome	0.985
Image Quality	Good, Fair, or Poor	0.363
Image Type	Photo or Illustration	0.928
Context	Advertisement, Cover Page, or Feature Story	0.974
Multiface	Yes or No	0.869
Race	American Indian, Asian, Black, Pacific Islander, White, or Unknown*	0.721
Smile	Yes or No	0.731

Table 1. Classification features and categories used for annotating facial images along with κ, quantifying the interrater reliability among three raters over 185 faces from the same magazine issue. * Denotes that this category was classified according to the current U.S. Census [About Race 2018]

3.2 Deployment of Web-Based Application

To scale up data collection, we created a web-based form in PHP, coupled to an SQL database, that could be deployed within crowdsourcing platforms to perform the two tasks required to obtain the data of interest. In Task 1, a magazine page was presented, and participants were instructed to crop any faces that are present; in Task 2, participants were instructed to categorize the faces identified in Task 1 according to the specifications in Table 1. The data collection protocol was to first complete Task 1 (cropping) on all our selected pages before moving on to the annotation phase, which allowed cropping errors to be eliminated before sending the extracted images for annotation. Task 1 was separated from Task 2 so that crowdsource workers would only have to be trained for and perform one scope of work. 13

While the data-collection interface is platform-independent and can be used to directly collect data, we found it beneficial to use AMT to recruit participants and manage payments. "Jobs" (or human interface tasks (HITs) in AMT vernacular) were deployed in AMT as a survey link. For Task 1, each job consisted of reviewing 50 pages and cropping all of the observed faces within each page. AMT workers were paid $5 USD (all payment rates cited here are in USD) for each completed job, which was based on the time it took student coders to complete similarly-sized jobs (30-40 minutes) with a goal of paying between $8-$10/hour, above U.S. federal minimum wage [Silberman et al. 2018]. For Task 2 jobs, AMT workers were required to categorize 25 to 50 faces, each of which was previously cropped from a page in Task 1. Student coders spent 10-15 minutes to complete jobs consisting of 50 faces on our context-free interface (discussed in section 3.4.1) and 15-20 minutes on jobs consisting of 25 faces on our default interface; therefore, AMT workers were paid $2.25 for these jobs. Once an assigned job was completed, the software generated a completion code that workers entered into AMT to receive payment. Using this code as an identifier, we were able to verify the quality of the work (see sections 3.3 and 3.4 for details) and process payments. For borderline or questionable work quality, we intentionally erred towards payment and only withheld payment for the most extreme circumstances. Each job also included an optional demographic survey, which will inform future studies exploring relationships between demographics and face annotation outcomes. All procedures were approved by the SUNY Polytechnic Internal Review Board. 14

3.3 Description of Task 1 (Cropping) Interface

In this task, workers were presented with a job consisting of 50 images, 47 of which were randomly-selected magazine pages and three of which were validation pages. On each assigned page, AMT workers were asked to crop a rectangle around individual faces by clicking and dragging from one corner of a rectangle to the opposite corner. (See Figure 1). If there was more than one face on the page, workers selected an option to remain on the page and continue cropping. Once all the faces were cropped, or if there were no faces on the page, workers selected an option to move onto the 15

next page in their job. We observed that workers often abandoned an assigned job after the first few pages, resulting incomplete jobs within our system. To eliminate these jobs, a script was created that ran in the background to look for pages that had been assigned within a job that had been inactive (i.e. no faces cropped) for more than 2 hours. Any data collected from these jobs was deleted and the pages within them were made available for a new job assignment.

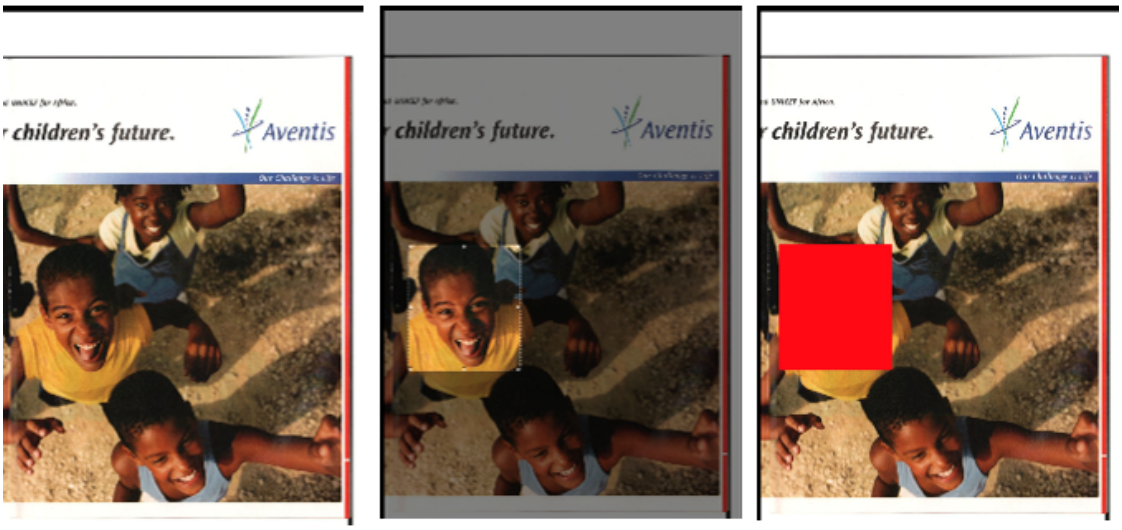

Figure 1. The cropping interface. Left: page as first presented to worker. Middle: worker selects face to crop. Right: Faces that have already been selected and submitted are covered up to help workers keep track of cropped faces.

Within each job, 3 of the 50 pages that the workers analyzed were validation pages, whose inclusion was meant to help detect workers that attempted to quickly receive payment by repeatedly indicating that there were no faces on each page, regardless of content. These pages were selected randomly from a database which contains a list of magazine pages and the known number of faces on each page, as determined by trained project personnel. These are our "ground-truth" faces. Worker quality was assessed by comparing the number of cropped faces on these pages to the known number of faces. Workers' validation page was flagged if they cropped more than one face on a validation page with only 1 face or cropped $\pm$ 1 face beyond the known number of faces on pages with > 1 face. When determining if payment should be provided, workers with 2 or 3 flags were subject to additional review while payment was immediately processed for all others.

Figure 2. Screenshot of a page within our in-house review interface. Selections cropped by workers are outlined with a red rectangle.

To facilitate the further inspection of AMT workers with a high number of flags, an easy-to-use, in-house review interface was built (Figure 2). On a single webpage, this interface displayed all of the magazine pages assigned to any worker, along with frames around the image areas that the worker selected for cropping. Using this interface, project personnel were able to rapidly scroll through the pages, inspect the work, and make note of pages with mistaken crops or faces left uncropped. If a worker had errors on more than half of their pages, then payment was not provided and all pages in their job were re-analyzed. We paid all other workers but used our revision process to identify pages with egregious errors, which were returned to the pool to have their analysis redone. 17

3.4 Description of Task 2 (Annotating)

In this task, workers were presented with a job consisting of either 25 or 50 images of faces, and were asked to enter appropriate tags for each face. The faces were randomly selected from the images that were cropped in Task 1. Procedures similar to those outlined in Task 1 were used to simultaneously manage multiple jobs, ensure that a sufficient number of images are available to populate each job, and cancel jobs that have timed out. For each face in a job, workers classified facial features according to the categories in Table 1 with an additional “not a face” option that served as a quality check for the collection of cropped faces. To maximize task efficiency, the options for each classification were presented as clickable radio buttons, rather than as drop-down menus. As in Task 1, once the job was completed, the workers were given a randomly-generated completion code that was used to secure payment through the AMT platform. 18

In a similar process to Task 1, each job contained 3 validation faces, also known as ground-truth faces, each of which

was consistently labeled the same by three student coders over all categories. To create a flagging system, we focused on the three categories that had the highest rates of agreement in our preliminary data collection: gender, image color, and image type. Magazine context had the second-highest interrater reliability, but as will be discussed in section 3.4.1, our software was configured to assess this feature in two different ways so it could not be used for validation. When the classifications matched the known values for a given validation image, the flag value was set to zero. Each mismatch contributed a value of 1 to the flag, with a maximum of 3. Images with large flag values were subject to further scrutiny. For the cases where an AMT worker had mismatches with the validation images, it was not possible to build a succinct visual inspection tool for all images as was done in Task 1 since category selections cannot easily be represented visually. Furthermore, there is a degree of subjectivity and ambiguity in certain categories, such as the presence of a smile, so we chose not to develop explicit criteria for processing AMT payments and all workers were paid. To navigate the potential for erroneous data and/or ambiguous categories, we obtained multiple annotations for each face, which were aggregated to obtain a crowdsourced label. As will be described in a subsequent section, we had each face annotated twice and resolved inconsistencies by choosing the label associated with the worker who was most consistent with other workers over all annotated faces, and who had the lowest number of flags.

3.4.1 Examination of variations in the interface

We also took this opportunity to examine how variations in the interface affected annotation results (see Figure 3). In particular, we were curious about whether faces taken out of context were more likely to be erroneously labeled. For example, a closely cropped face may not include gender cues, such as hair and clothing. To address this question, we developed two different annotation interfaces. In the context-free version, we show only a cropped face to workers, who then determine the characteristics. Because there is no context around the face, the magazine context (ad, feature, cover, etc.) and multi-face (whether the face being tagged is accompanied by other faces) categories were required to be determined in Task 1 while workers did the cropping. In the second (default) version of the task, workers see the full page with a rectangle around the face of interest when labeling the face and workers answer questions about the face as well as about the context around it. We default to this later version of the interface since we were able to automate Task 1 (see section 5.2), requiring the context annotations to be assigned in Task 2. We found that, despite there being only two additional questions in the default version of the interface compared to the context-free version, it took almost twice as long to complete the labeling tasks, which is why AMT jobs consisted of 25 rather than 50 faces with the default version.

Figure 3. Left: The context-free version of the interface shows workers only the face to be annotated. Right: The default version of the interface shows the full page with the face in question outlined in green.

4. Software Evaluation: Case Study with Magazine Archive

A case study was performed using a subset of our magazine archive consisting of one July issue selected from every year between 1961 and 1991, which corresponds with our historic period of interest. Additionally, each of the one-per-decade issues that the student coders manually labeled during our preliminary studies were used as a second data set. The first data set was denoted as 30YR (it spans 30 years) while the second was called OPD (as we selected One issue Per Decade). After being cropped, both the 30YR and OPD data were each labeled by two distinct AMT workers. 21

4.1 Summary of AMT Accuracy

A total of 87 AMT workers cropped 3,722 total pages in Task 1. Due to various glitches that were discovered during deployment and eventually rectified, certain jobs contained less than 50 pages with the average being 47.18 pages per job. The average time to complete a job was 47 minutes. Three validation pages were randomly included within each job to address concerns about individuals incorrectly indicating there were no faces on a given page. However, this behavior was not widely observed, as less than 5% of all validation pages were characterized as having no faces. More common errors appear to have been cropping only a fraction of the faces present on a given page or including many faces within a single crop. For example, 20.0% of validation pages with 3 or more ground truth faces were characterized as having only 1 face. The cropping error rate was significantly reduced when workers were required to acknowledge that they read our instructions before beginning the job. Overall, for 72.8% of validation pages, the number of faces identified by the AMT workers agreed with known number of faces. For an additional 7.6% of validation pages, AMT workers cropped more faces than the known number. It is likely that these cases represent genuine attempts at completing the task, where the known faces along with additional small, poor quality faces were cropped. Processes were implemented to eliminate poor quality faces (see section 4.3). Therefore, the cropping accuracy should consider true positives to be those validation pages where the number of cropped faces either matched or exceeded the ground truth, which led to an effective accuracy of 80.4%. Each page was verified with our inspection interface described above and crop errors were corrected before proceeding to Task 2. 22

In Task 2, a total of 342 workers annotated 9,369 faces. One AMT assignment consisted of either 25 or 50 faces, depending on whether the default or context-free interface was being used. Technical glitches, which were later 23

corrected, occasionally caused the number of faces in a job to slightly vary. The average time to complete a job was 30 minutes using the context-free interface, with a job consisting of 50 faces, and 25 minutes using the default interface with a job consisting of 25 faces. Table 2 illustrates the consistency of image annotations with the known labels of the validation images. With the exception of image quality, the accuracy for each category was above 87%.

Photo	Color	Angle	Quality	Gender	Race	Smile	Adult
0.96	0.93	0.88	0.48	0.94	0.88	0.89	0.99

Table 2. Proportion of images where AMT worker's label matched the known validation image label in Task 2. Results are not provided for Magazine Context or Multi face since these categories were assessed in Task 1 when the context-free interface was used.

4.2 Comparison of Default versus Context-Free Interface for Task 2

As described in section 3.4.1, Task 2 was deployed with two different interfaces. In the default case, faces were presented in the context of the original page they were cropped from, while in the context-free case, the face alone was presented. To investigate whether the interfaces affected the labeling task, we used the default interface for both rounds of OPD labeling, but varied the interface for the 30YR data, as shown in Table 3. We then examined the consistency of labels over these two cases. 24

	Round 1	Round 2
OPD	Default Interface	Default Interface
30YR	Context-free interface	Default interface

Table 3. Deployment of Task 2 over various interfaces.

For each of the 10 labeled features, the proportion of images where the ratings agreed was calculated for both the 30YR and OPD data sets. The results are illustrated in Table 4. According to χ^2 analyses, the differences between the proportion of matches was significant for 5 of the 10 features with the largest differences being between magazine context and image quality. This is to be expected since the two different interfaces used for the 30YR data primarily differed in ways that can be expected to affect these features. There were relatively large differences in image quality based on the presence of context, with 54.1% and 9.6% of faces labeled as good and poor quality, respectively, compared to 3.4% and 20.5% of images labeled good and poor, respectively, for the context-free design. It is possible that the presence of context increased the readability of the face. 25

Interestingly, the correspondence in magazine context was larger across different interfaces in the 30YR data than across the consistent interfaces in the OPD data. The observed statistically significant differences may be due to the large sample size, which is bolstered by effect sizes (Cohen's *f*) that are well below 0.1 in every case; typically, a moderate effect is considered 0.3. As a result, we conclude that the differences in annotation quality according to the interface design are relatively small. 26

	Multiface	Color	Context	Photo	Angle
30YR	0.68	0.90	0.69	0.92	0.79
OPD	0.74	0.88	0.60	0.91	0.78
p	0.004*	0.11	<0.001*	0.48	0.47
effect	0.04	0.02	0.06	0.01	0.01
	Gender	Race	Adult	Smile	Quality
30YR	0.90	0.71	0.93	0.81	0.45
OPD	0.89	0.77	0.97	0.82	0.53
p	0.18	0.002*	0.001*	0.72	<0.001*
effect	0.02	0.05	0.05	0.005	0.05

Table 4. Proportion of ratings agreeing for both 30YR and OPD data with χ^2 analysis *p*-values and effect sizes (Cohen's *f*) for the differences in proportions provided. * indicates a p-value < 0.05.

4.3 Effect of Image Quality

We next explored the effect of image quality on the consistency between raters. Each image was classified as having *Satisfactory Quality* (SQ) if both raters scored its quality as either good or fair, or *Non-Satisfactory Quality* (NSQ) otherwise. Approximately 27% of the observations were classified as NSQ. The proportion of matches for each feature was then calculated separately for both the SQ and NSQ cases. The results are illustrated in Table 5. For 6 of the 10 features, χ^2 analyses indicated that the concordance between raters was significantly different for SQ and NSQ images. The effect sizes (Cohen's *f*) were larger than when comparing 30YR to OPD images with the adult and image quality features approaching a moderate effect. 27

	Multi-face	Color	Context	Photo	Angle
SQ	0.68	0.90	0.68	0.94	0.81
NSQ	0.71	0.89	0.68	0.85	0.75
p	0.11	0.11	0.60	<0.001*	<0.0001*
effect	0.02	0.02	0.008	0.14	0.06
	Gender	Race	Adult	Smile	Quality
SQ	0.94	0.75	0.97	0.82	0.56
NSQ	0.81	0.64	0.85	0.81	0.21
p	<0.001*	<0.001*	<0.001*	0.52	<0.001*
effect	0.18	0.11	0.21	0.01	0.31

Table 5. Proportion of ratings agreeing for both SQ and NSQ data χ^2 analysis *p*-values and effect sizes (Cohen's *f*) for the differences in proportions provided. * indicates a p-value < 0.05.

The results in Table 5 indicate that it may be advantageous to eliminate NSQ data from subsequent analyses. Before doing so, it is important to determine if this will introduce a bias. Due to changes in printing technology and subject matter over the 90+ years spanned by the data, there is the potential for image quality to differ by time. This possibility was assessed by separately calculating the frequency of SQ and NSQ images in each issue. A χ^2 analysis was then performed, which indicated that there was no significant difference between the SQ and NSQ frequency distributions. Therefore, eliminating the NSQ images will not introduce temporal bias. 28

4.4 Aggregation of Multiple Image Labels

Each face was annotated twice, each time by distinct AMT workers. While the majority of labels (~ 80%) were in agreement, we required a methodology to resolve disagreements between labels in order to have a definitive value for each annotation. When crowdsourcing data, this is often achieved by having multiple individuals rate a given image and then using a majority rules approach for each feature [Hipp et al. 2013] [Hipp et al. 2015] [Nowak and Rüger 2010]. However, this approach can be resource intensive. More targeted approaches have been developed that implement an expectation-maximization algorithm to determine the most likely label for a given object in order to ultimately determine a score for the quality of each work [Dawid and Skene 1979] [Wang et al. 2011] [Organisciak et al. 2012] [Welinder and Perona 2010] [Whitehill et al. 2009]. Lower-performing workers can then be filtered out of the rating system. We aimed to emulate such approaches, but with a simplified procedure that functions over only two coders per image. Our strategy was to calculate a proficiency score for each of the raters and to resolve inconsistencies by selecting the response recorded by the individual with the better proficiency score. Proficiency scores were determined for each worker by examining their validation images and calculating the fraction of annotations matched between the worker's input and the ground truth. A proficiency score of 1 is a perfect score. The average proficiency score (μ) was 0.87 with a standard deviation (SD) of 0.09. An alternate way to calculate the proficiency score was by considering all of the images tagged by a given rater and computing the average fraction of image features that matched the images' other raters. The average proficiency score with this convention was μ = 0.81 with SD = 0.06.

29

Flag Sum	0	1	2	3
Mean Proficiency (All Rated Image)	0.82	0.80	0.78	0.64
Mean Proficiency (Validation Images)	0.90	0.85	0.77	0.76

Table 6. Mean proficiency score stratified by the sum of flag values over all validation images.

Table 6 compares our two methods of calculating the proficiency score with the flagging system for image annotations. The sum of the flags for each participant was calculated and proficiency scores were stratified by these values. As shown in Table 6, lower proficiency scores were associated with larger flag values, which indicates that our flagging system provides a reasonably good indicator of worker proficiency. An ANOVA test indicated that the differences in proficiency score values among the flag values were significant ($p<0.001$) for both varieties of the proficiency score.

30

4.5 Validation of Proficiency Score

Prior to deploying the proficiency score methodology to resolve annotation inconsistencies throughout the entire corpus, it was necessary to determine the consistency of this methodology with the more established majority-rules procedure. To assess this, a subset of 1,000 SQ images were selected from the corpus at random and then submitted to AMT for three additional annotations (i.e., five total annotations). The annotation label selected most frequently was selected for this image with ties between annotation labels (< 1% of all annotations) chosen at random. Table 7 summarizes the proportion of faces for which the annotation labels in the five-rater consensus and proficiency score (using the all rated images option) matched. These results indicate that the proficiency scoring procedure is sufficiently accurate to allow future iterations of this system to proceed with only two raters per image, which will allow for a more resource-efficient project.

31

Photo	Color	Angle	Quality	Gender	Race	Smile	Adult	Context	Multiface
0.97	0.97	0.92	0.74	0.97	0.93	0.91	0.99	0.90	0.85

Table 7. Proportion of images where the five-rater consensus and proficiency score labels matched, stratified by annotation category.

5. Software Applications

5.1 Applying Software to Other Data Sets

While this software was built for our specific purpose of cropping and annotating faces from *Time* magazine, we were mindful about its generalizability and developed it with the hope that it could serve as a useful tool for other researchers with other corpora. To this end, the code is hosted on GitHub (https://github.com/Culture-Analytics-Research-Group/Data-Collection) and is written so that both tasks (image cropping and annotation) are easily generalized, and the annotation variables are straightforward to modify. Instructions for modifying this software to a different archive, along with detailed instructions on how to use the software, are provided in the Appendix. 32

As a demonstration of this flexibility, we hosted a proof-of-concept workshop in December 2018 demonstrating the use of our tool on selected pages from the GQ Magazine corpus [Jofre et al. 2018]. Prior to the workshop, we used our trained face detector to identify and crop faces from these pages, and the workshop demonstrated Task 2 (annotating the selected images) to explore trends in facial hair. 33

The cropping part of the software (Task 1) is particularly easy to adapt for cropping other objects. In our own research, we are currently using the cropping part of the software to extract the advertisements from the corpus. The software is also being used to identify measures of neighborhood distress (graffiti, abandoned vehicles, etc.) in a study that examines the role of environmental factors in promoting physical activity. 34

5.2 Task Automation

Our case-study data has provided us with a corpus-specific training set that we have used to train a RetinaNet detector [Lin et al. 2017] [Lin et al. 2018] to automatically identify and extract the rest of the faces from the archive [Jofre et al. 2020a] [Jofre et al. 2020b]. Our case-study data set of 1,958 pages with 4739 face annotations and 1708 pages containing zero faces was used to train the detector. The detector was trained for twenty epochs, since training for more resulted in overfitting and poor generalization in face detection across different historical eras. After running the detector on every page from the archive over 400 thousand facial images were extracted, using a threshold of 50% certainty. When we increased the accuracy threshold to 90%, we were able to extract over 327 thousand faces with very high accuracy. In comparison, our first attempts at automated extraction with OpenCV yielded only 117 thousand facial images from the entire corpus, and 5% of these were false positive (i.e. not actually faces). Compared to OpenCV, the trained RetinaNet detector was able to extract more faces, particularly those with a profile orientation, and those that were illustrated instead of photographed. 35

We have also trained classifiers to automatically label the gender of the face by fine-tuning a pre-trained VGG Face CNN Descriptor network [Parkhi et al. 2015] [Malli et al. 2018] with our crowd-sourced data. From the initial set of data described here, 3,274 faces were male, and only 1,131 were female, which skewed our results on the first run. To expand the training set, we employed a bootstrapping technique to acquire additional, more balanced, training data and thus improve our classifier. The model trained on the AMT data was used to classify all 327,322 faces from the archive. From these faces, we randomly selected images and manually verified the classification results. These new images plus the AMT data yielded a new dataset of 17,698 faces for the second round of training, with roughly equal male/female representation. This yielded a 95% accuracy [Jofre et al. 2020a] [Jofre et al. 2020b]. 36

5.3 Visualizing Annotation Results

We created an additional piece of software, also available on our Github page (https://github.com/Culture-Analytics-Research-Group/Metadata-Analysis), that pulls the data directly from the database where the crowdsourced annotations are stored and creates visual summaries of image annotations versus time. The user can select any annotation category and easily generate a chart of the selection as a function of time, aggregated by year or by month. In addition, the tool allows users to select subsets of categories. The example in Figure 4 shows the percentage of women's faces out of the subset of faces identified in the context of advertisements. This tool is intended for preliminary analysis that allows researchers to quickly identify temporal trends and patterns. 37

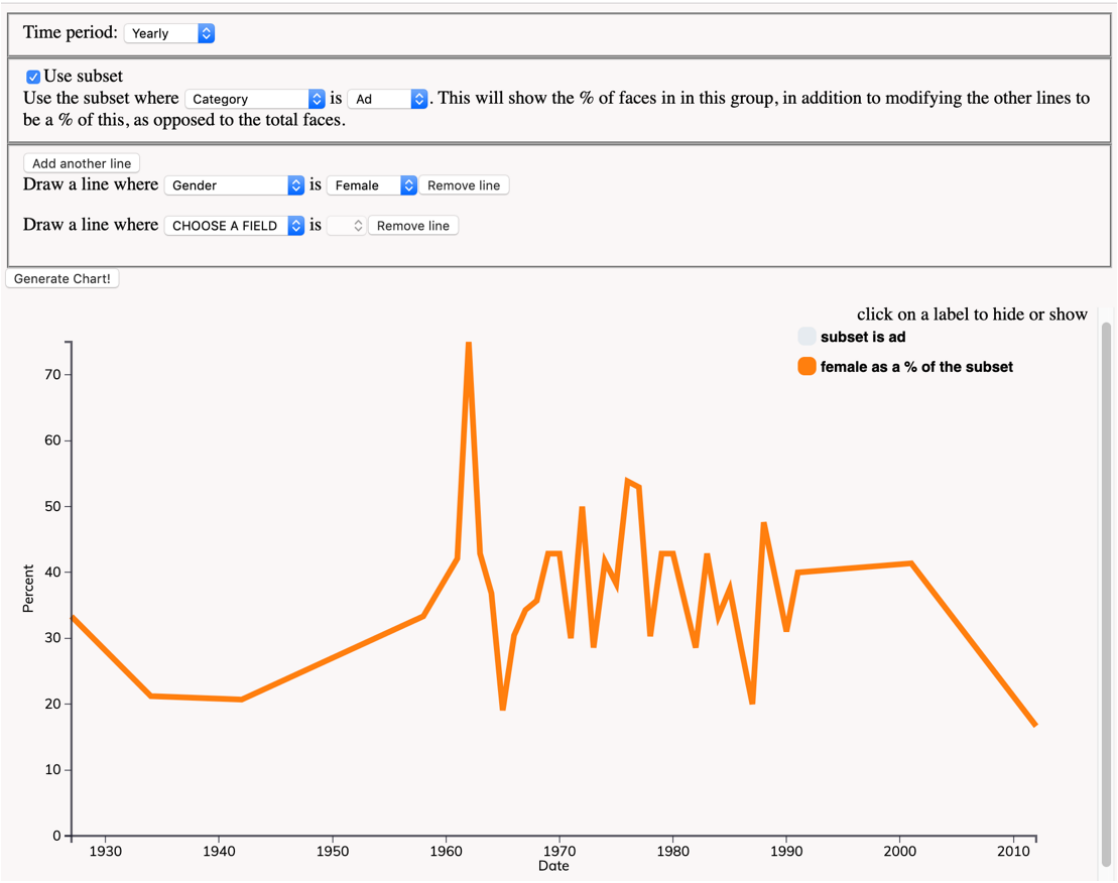

Figure 4. Screenshot showing the percentage of faces that are tagged female out of faces that are tagged as being within advertisements.

5.4 Digital humanities studies

The data we collected with these methods have allowed us to generate more data via machine learning, and has allowed us to ask the following questions [Jofre et al. 2020a]. How has the importance of the image of the face changed over time? How has gender representation changed over time? How does gender representation correlate with the magazine's text and with the historical context? How has race representation changed over time? How has the representation of children changed over time? How does race and/or age correlate with the magazine's text and with the historical context? What types of faces are more likely to be smiling? In what context (ads or news) do certain types of faces tend to appear, and how does this change over time? What types of faces are more likely to be presented as individualized portraits? 38

In our own work, we used the data collected through this method (as well as the automatically-extracted data that this work made possible) to examine how the percentage of female faces found in *Time* magazine between the 1940s and 1990s correlates with changing attitudes towards women. We found that the percentage of images of women's faces peaks during eras when women have been more active in public life, and wanes in eras of backlash against women's rights. The changes in the representation of women in the magazine over time tracked closely not only with the overall historical context, but also with the internal policies of the publication, and with a close reading of the magazine's content. We believe that this finding is particularly relevant in our contemporary post-literate world in which people absorb culture primarily through images [Jofre et al. 2020b]. 39

6. Discussion and Future work

6.1 Observations

We were successful in building and deploying software to manage the crowdsourced extraction and labeling of features 40

from an image-heavy corpus. While the software is generalizable, we focused on an application where faces were required to be extracted and labeled from *Time* magazine. The accuracy for both Task 1 and Task 2 were in line with those seen for other studies that have used crowdsourcing for similar tasks [Hipp et al. 2013] [Hipp et al. 2015] [Nowak and Rüger 2010]. In contrast to these other studies that required multiple workers for each image, our method only requires two individuals to annotate each image to gain results with a similar accuracy.

Our case-study results show that the differences between labeling performed on context-free versus context-rich interfaces were small. However, there was a notable difference when we instead compared images that were tagged as "good" quality with images tagged as "poor" quality, an effect likely due to challenges in reading poor quality figures. This indicates that there is value in requiring workers to evaluate image quality, as it allows us to flag potentially ambiguous annotations. Interestingly, faces that were viewed in the context of a full image were less likely to be labeled as having poor quality compared to faces that were viewed in the context-free interface. It seems that context increases the readability of the face in question, which makes our default interface advantageous. On the other hand, a disadvantage of the default interface is that it takes nearly twice as long to label a single face compared to the context-free interface. While the default interface contains two additional features to be assessed, we speculate that providing a full image rather than a cropped image adds a significant cognitive load to the task. We anecdotally note that personnel who tested both interfaces observed that the default interface felt "less tedious" than the context-free interface: viewing pages from vintage magazines was "more entertaining" than viewing decontextualized images of faces. In the end, we likely will opt for the default interface in our future studies. This is in part because we have been able to fully automate image extraction, but also because the context-rich environment seems to increase the readability of the selected face. An image of a face alone loses the rich contextual information of the complete page in which it appeared. 41

Using the methods described in this case study, we successfully collected data that was 1) used to train an object detector and an image classifier, 2) published and made accessible to other digital humanities researchers [Jofre et al. 2020a], and 3) used to undertake a study on gender representation in *Time* magazine [Jofre et al. 2020b]. 42

6.2 Advantages of a Standalone Application

While AMT offers multiple options, including developer tools and a sandbox, for creating image cropping and tagging interfaces, we chose to build our own web-based application for several reasons. For one, this allows complete customizability, which was beneficial as we tweaked our approach in response to preliminary data. Also, this web-form enables us to collect data in a manner that is independent of any service providers, which allows us to use different services without compromising our methods. In this work, we used AMT to provide a proof-of-principle, but we plan to deploy this system on other crowdsourcing platforms. The stand-alone interface also opens the possibility of collecting data with volunteer crowdsourcing, as has been done in projects from the New York City Public Library [NYPL Map Warper 2018] [NYPL Labs 2018] [All Hands on Deck 2018]. The biggest challenge in using volunteers is generating sufficient interest to collect a significant amount of data. We may have to consider methods of gamifying the tasks to make them more appealing, and our hope is that once our results are presented publicly, people may become interested in participating in the project. Lastly, a standalone application can be shared with other researchers and adapted to different types of projects in a way that is not possible with platform-specific approaches. 43

6.3 Limitations

From a humanistic perspective, there is a limitation in using only visual data to classify race and gender. In the case of gender, our data doesn't distinguish between someone who identifies as a woman (or man) and someone who presents as female (or male), and the automatic classification trained on this data assumes that gender is binary, which is problematic. Human coders, who see the context of the page can mitigate this problem by labeling the gender as 'unknown', which accounted for 6% of the faces. However, upon closer inspection, we found that none of these were actually gender non-binary adult individuals: many were not faces at all (errors in the face extraction), many were very small low resolution images that were hard to read, some were non-gendered cartoon illustrations (a face drawn onto an object, for example), and some were infants or small children. So, while problematic, the assumption of a binary gender may be suitable for examining certain mainstream 20th century publications such as *Time* magazine. In the case of 44

race, we found its classification was difficult because race categories are somewhat arbitrary, and because the concept of race is highly context-dependent. Census categories have changed significantly over the past century and they continue to be contentious. In our experience with human coders, we found that the race of a face is often not recognized unless it is embedded within a stereotyped setting, and that when the face was not white, coders tended to disagree on race more than with other categories.

A second, more practical, limitation is that this software requires that the user have some familiarity with PHP and with managing SQL databases. Our goal was to make a useful tool for researchers, rather than a polished commercial product. Researchers using this software need to have someone on their team with basic programming experience. The tradeoff, however, is that this software allows researchers to have full control of the data collection and quality controls. 45

6.4 Long term project goals

Our next steps are to continue using this crowdsourced data we collected to automate the classification of other categories, and to undertake a close examination of the context in which faces appear, particularly advertisements. To this end, we are using our software to crowdsource the extraction of all advertisements from selected issues of the corpus. These will be used to train an algorithm that will extract all the advertisements from the corpus. Using this advertising data in conjunction with our face data will allow us to undertake a study on trends in advertising in this particular media outlet. 46

The ultimate goal of this project is to create web-based interactive visualizations of the data we extract from our *Time* magazine archive, and of the results of our analysis. We hope to provide insights into how depictions of faces have changed over time and what such changes in visual representation can tell us about the intersection of politics, culture, race, gender, and class over time. We hope that the online resource we create will be of interest to researchers and students of media and cultural history, as well as to the general public. Our visualization approach is inspired by Manovich's Selfie-city and Photo-trails work [Manovich et al. 2016] [Douglass et al. 2011] [Hochman et al. 2016], and by his team's use of direct visualization [Crockett 2016], which is an effective way to engage broad audiences into complex corpuses. We also draw inspiration from Robots Reading Vogue [King and Leonard 2016] and Neural Neighbors [Leonard and Duhaime 2018], which are projects based in the Yale University library system. Most recently, we have been using and modifying software from Yale's DH lab, *PixPlot* [Duhaime 2018], to sort the images with unsupervised clustering. 47

In addition to gaining insights from our corpus and making these publicly accessible, we also aim to develop novel methodologies for the visual analytics of large, image-based data sets that can be applied to a variety of projects and shared with other researchers. 48

Acknowledgements

We would like to acknowledge Michael Reale for his help with automating image extraction and tagging. We would also like to acknowledge generous research support from our institutions, SUNY Polytechnic and Chapman University, for the start-up funding that made this research possible. Finally, we acknowledge IPAM at UCLA for bringing this collaboration together at the Culture Analytics Long Program and for equipping us with the tools to undertake this research. 49

Appendix: Using and modifying the software

Part 1: Details About the Code

This is a web interface for gathering data from images on a large scale. Users should serve it with accompanying writable SQL databases. We provide the accompanying database structures here and on Github, along with the code. 50

This web-based interface facilitates gathering data from images: it allows users to crop a selection from a larger image and to input information about the crop. In our case, we are selecting faces out of images from a magazine archive, but 51

with some minor edits this code can be used to select anything else from an image archive (cars, trains, signs, etc.).

This web interface is platform independent. Users only need a link to access it.

The code itself has three different data gathering surveys that are part of it.

The first survey allows participants to select and save a cropped portion of an image. The survey contains multiple pages (in our case 50), and the participant has to select and submit all the faces from each page. To access the cropping survey use the link “survey.php?load=crop”.

For the crop, we used https://github.com/odyniec/imgareaselect “imgareaselect” by Michal Wojciechowski.

The second survey allows users to classify the already cropped images from a selection of categories. To access the cropping survey use the link “survey.php?load=tag”.

The third survey is simply a demographics survey that allows users to enter their demographic information, and is presented at the end of each of the previous two surveys.

The code of this survey is split into 4 different files *instructions.php*, *survey.php*, *post.php*, and *functions.php*.

instructions.php is a landing page that presents the user with instructions for the current survey either the cropping survey or the classify survey. The survey and instructions that will be presented are determined by the GET variable load in the URL. If load=crop the crop instructions are presented if load=tag then the classifying survey is presented. Users must select that they have read the instructions in order to move onto the survey.

survey.php is the main interface of the survey that the user interacts with.

If the job is to crop images, the url “survey.php?load=crop” should be used. The image to be cropped is presented and users are asked if the object to be cropped is present (faces in the case of the original purpose) in the image. If the object is present users can crop it be clicking and dragging over the object in the image. If multiple objects are present users may select that there are more objects (faces) on the page. Any previous cropped objects will be covered when cropping another object. If it is not present users may simply select that the object is not there and move to the next image.

If the job is classifying images that were previously cropped, the url “survey.php?load=tag” should be used. The user is presented the image from which an object of interest was cropped, with the cropped portion highlighted along with questions about the classification of the object.

Each job within the survey has a total number of images to be done at one time that can be set along with three check points that can be set (in *functions.php*). The check points present the user with ground truth pages where the classification or number of objects cropped is already known in order to check whether a user has properly completed the survey. These variables can be set in *functions.php*.

post.php handles all submission of data to the data base after a user has hit the submit button. If the job was cropping, data is submitted to the database and the selected portion is cropped and saved to a folder on the server. If the job was classifying, data is just submitted to the database. If a user has completed a check page then information on the page is placed in an array to later be checked and entered at the end of the survey. If the user has reached the end of the survey and filled out the demographics information then the demographics data and check data is submitted and a completion code is generated. If a user has no activity for 2 hours and then tries to submit data *post.php* will cause the session to timeout.

functions.php contains all the functions that are used in the survey and is included in both *survey.php* and *post.php*

functions.php Overview

$job — php $_GET variable that indicates whether the job I for cropping or tagging so that the proper page is loaded.

Obtained from the url, for example, in the url “survey.php?load=crop” $job=crop.

$batch_size — variable controlling the number of images per job 67

$check — array variable that contains when ground truth images will be shown in the job 68

$face_total — variable for cropping that keeps track of the number of objects cropped from a specific image 69

$file_array — holds image file names to have a group number added at the end of each job 70

$check_data1, $check_data2, $check_data3 — holds data submitted by users on each of the three ground truth images 71

db_connect() — returns a mysqli_connection object for connecting to the database, set $servername, $username, $password, and $database you wish to connect to 72

select($job, $batch_size, $connection) — selects images one at a time as long as there is enough images available for another job, otherwise users are presented with a message that requests are currently at capacity. This function also marks pages as being worked on in the database and adds a timestamp for clearing data on a job that was never finished. The file name of the image is returned 73

check_select($job, $connection) — similar to select, except it selects ground truth images from their tables. 74

parse_filename($job, $filename) — parses information from the file name of the image. If the job is cropping, then this information is used to create the path that cropped images will be stored in. If the job is classifying, then this information is used to determine the path of the original image. The parsed data is stored in the $file_data array to later be displayed and submitted to the database. This function is based on the file name scheme of the images originally used with this code. 75

display($job, $file_data) — handles what is displayed for the user depending what the job is. Inputs for the survey questions are printed out as radio buttons 76

hidden($job, $batch_current, $filename, $file_data, $file_array, $check_data1, $check_data2, $check_data3) — prints out the hidden inputs for each job mainly the data parsed from the filename. If the job is cropping the hidden inputs containing information for cropping the data is printed out. 77

post_hidden() — prints out hidden inputs for *post.php* that need to be sent back to *survey.php* 78

crop_image() — handles the cropping of images for the crop job and accounts for offset of different window resolutions and sizes. 79

post_variables($job) — sets the variables in post that will be submitted to the database for each job along with variables needed for post functions 80

submit($job, $connection) — submits data to the database for each job and marks images as no longer being worked on. If the job is cropping and no object was cropped then no data is submitted. If the job was cropping and the page was a ground truth page a temporary entry is mad in a table so that covering previously cropped objects on pages with multiple objects will work properly. 81

final_submit($job, $connection) — submits the demographics information to the database. A group number is generated by selecting the highest group number from the database group tables for each job and adding one. 82

This group number is assigned to each image that was part of the job. It is also inserted into the check table for each job along with possible flags raised from the information in the check arrays and a randomly generated code that will be presented to the user. This code is for admins to manage payment via Amazon Mechanical Turk. 83

demographic($job, $file_array, $check_data1, $check_data2, $check_data3)- displays the form and the inputs for users 84

to enter their demographic information

coverfaces($job, $connection, $filename, $file_data) — 85

If the job is set to “crop”, covers previously cropped objects (faces) on images where multiple objects need to be cropped, by selecting previously submitted x and y coordinates from the database. If the image is a ground truth image then it selects from the temporary entry in the table for crop checks. If the job is set to “tag”, this function is used to find the coordinates and draw the rectangle around the object to be classified. 86

Part 2: The Data Tables

Below is the “pages” table structure — Used for the cropping task. 87

Column	Description
“page_file”	File name of magazine page image
“faces”	Total number of faces on that page (starts out as null until page is analyzed)
“group_num”	Identifies a completed job. This cell is null until a job is completed, when the job is completed, all the pages that belonged to that job are marked with this group number. This number is unique and increments each time a job is completed.
“working”	flags whether that particular page is being worked on by another worker.
“timestamp”	which marks the date/time a page is displayed. If a page was displayed more than 2 hours ago and does not have an associated “group number”, then any data collected on that page is cleared, timestamp is marked null, and the page is made available again for selection.

Table 8.

Below is the “crop_groups” table structure — Used to track workers in cropping task. 88

Column	Description
“group_num”	Job identifier
“flag1”	Results from “ground-truth” comparisons.
“flag2”	
“flag3”	
“code”	Unique completion code. Randomly generated by our software, to be entered into mechanical turk.

Table 9.

Below is the “ground_truth_crops” table structure. This is the ground truth table that is used for the cropping task. 89

Column	Description
“file”	File name of the image
“nfaces”	Number of faces on this image
“working”	Marks whether the file is currently being used
“timestamp”	Marks time that file was displayed. Resets after 1 hour.

Table 10.

Below is the “tag_groups” table structure – Used to track workers in tagging task. 90

Column	Description
“tag_group”	Job identifier
“flag1”	Results from “ground-truth” comparisons.
“flag2”	
“flag3”	
“code”	Unique completion code. Randomly generated by our software, to be entered into mechanical turk.

Table 11.

Below is the “data” table structure — this is the table that contains the collected data. Year, month, day, page, image, and coordinates are populated during the cropping task. The rest of the columns are populated in the tagging task. 91

Column	Description
“year”	These identify the source image, which is labeled by issue date and page number.
“month”	
“day”	
“page”	
“multiface”	Is there more than one person in the image (yes/no)?
“category”	Is the image part of a feature story, an ad, or the cover page?
“color”	Is the image in color or monochrome?
“photo”	Is the face a photograph or an illustration?
“angle”	Is the face in profile or looking straight ahead?
“gender”	Is the face male or female (or other)?
“race”	What is the race of the face? (select from 5 census categories: White, Black, Asian, American Indian, Pacific Islander)
“adult”	Is it an adult or a child?
“smile”	Is the face smiling?
“quality”	What is the image quality like? (Good — face is clearly visible, Fair — face is small or slightly blurry, Poor — face is barely visible, Discard — this is not a human face)
“image”	the name of the cropped image that is saved in the data folder on the back end.
“x1”	These are the diagonal corner coordinates of the cropped selection.
“y1”	
“x2”	
“y2”	
“tag_group”	tracks completed tagging jobs. This cell is null until a job is completed, when the job is completed, all the crops that belonged to that job are marked with this group number. This number is unique and increments each time a job is completed.
“working”	, “working” flags whether that crop is currently being tagged by another worker
“timestamp”	, and “timestamp” marks the date/time an object is displayed for tagging. If an image was displayed more than 2 hours ago and does not have an associated “tag_group”, then any data collected on that crop is cleared and the crop is made available again for selection.

Table 12.

The “ground_truth” table has the same structure as the data table — This is the ground truth table for the tagging task. 92

The “crop_check” table stores the year, month, day, page, and coordinates of the ground truth pages that the user crops. This keeps track of the objects cropped out of the “ground truth” pages. It is used to cover objects that a user has already cropped from a single page when multiple objects are present, and it is used to calculate the flags in the 93

“crop_groups” table. Once the job is finished and the flags are calculated, the entries in this table are deleted.

“tag_check” table structure (this table records workers’ entries on the validation pages) 94

Column	Description
“tag_group”	Job identifier
“multiface”	Is there more than one person in the image (yes/no)?
“category”	Is the image part of a feature story, an ad, or the cover page?
“color”	Is the image in color or monochrome?
“photo”	Is the face a photograph or an illustration?
“angle”	Is the face in profile or looking straight ahead?
“gender”	Is the face male or female (or other)?
“race”	What is the race of the face? (select from 5 census categories: White, Black, Asian, American Indian, Pacific Islander)
“adult”	Is it an adult or a child?
“smile”	Is the face smiling?
“image”	The validation image used

Table 13.

Part 3: Instructions for Modifying the Software for Use in Other Studies

While this software was built for our specific purpose of cropping and annotating faces from a specific periodical archive, we were mindful about its generalizability and developed it with the hope that it could serve as a useful tool for other researchers. We share our code and database structure on GitHub with this intent. The code is written so that the cropping job is easily generalized and the annotation variables are easy to modify. 95

The most straightforward application of this software is for researchers interested in cropping and annotating objects from other magazine archives. To use our application, the archive needs to be stored as a collection of .jpg images named using the following convention: YYYY-MM-DD page X.jpg (where YYYY is the year, MM is the month, DD is the day, X is the page number). We share the database structure so that users can easily configure it from their server. Users can change column names (and corresponding variable names in the code) as needed. 96

The key part of the code consists of four php files: *instructions.php* is a landing page in case users want to present workers with instructions at the beginning of a task, *survey.php* contains the interface the worker interacts with, *post.php* handles all the submission of data to the database, and *functions.php* contains all the functions used in *survey.php* and *post.php*. The user will have to modify these files, depending on the application. At a minimum, the user will need to edit the *db_connect()* function in the *functions.php* file with their own server configurations. 97

To use the cropping task, users should list the images they want analyzed in the *page_file* column in the *pages* data table and serve the ‘survey.php?load=crop’ URL to display the cropping task. (A link to the demo will be included here if this paper is accepted, after anonymity is lifted.) In the *function.php* file, users can adjust the number of pages that comprise a job, the number of validation images per job, and the location of the validation images (2nd image seen, 5th image seen, etc.). The validation images are drawn from the *ground_truth_crop* table, which the user must populate. 98

When a worker crops a face with this interface, a copy of the cropped image is stored on the backend and the *data* table is populated with information about this face. The user must specify the name and path of the folder where the cropped images will be stored: this is done in the *crop_image* function in *functions.php*. The information stored in the data table is the year, month, day, and page number, parsed from the source image name; the coordinates of the crop; and the name of the cropped image. If users need to have a different file naming convention and need their source image names parsed differently, they can modify the *parse_filename()* function in the *functions.php* file. The total number of crops made per page is stored in the *faces* column of the *pages* table. If the user is cropping an object other than a 99

face, the names of variables, data columns, and the descriptors on the frontend can be changed to more appropriate terms.

To display the annotation task, users should serve the “survey.php?load=tag” URL. (a demo page can be viewed here: https://magazineproject.org/TIMEvault/survey.php?load=tag .) To use the tagging task, the *data* table should be populated with the source image identifiers (year, month, day, and page) and with the coordinates of the crop. If the user wants to use the context-free version of the interface, they will only need to provide the name of the cropped image in the *data* table and modify the source image in the “content-div” html element in the *survey.php* file.

If users want to annotate features that are different from the ones we listed, the names of the data columns can be changed, as well as the corresponding variable names in the functions *post_variables()*, *submit()*, and *display()*, which are in the *functions.php* file. Data columns and corresponding variables can be added or removed as needed.

Works Cited

About Race 2018 About Race, United States Census Bureau. URL https://www.census.gov/topics/population/race/about.html (accessed 9.20.18).

All Hands on Deck 2018 “All Hands on Deck: NYPL Turns to the Crowd to Develop Digital Collections”. *The New York Public Library*. URL https://www.nypl.org/blog/2011/09/15/all-hands-deck-nypl-turns-crowd-develop-digital-collections (accessed 9.6.18).

Bradski 2000 Bradski, G. “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools* (2000).

Caltech-UCSD Birds 200 2018 Caltech-UCSD Birds 200. URL http://www.vision.caltech.edu/visipedia/CUB-200.html (accessed 9.18.18).

Casey et al. 2017 Casey, L. S., Chandler, J., Levine, A. S., Proctor, A., Strolovitch, D.Z. “Intertemporal Differences Among MTurk Workers: Time-Based Sample Variations and Implications for Online Data Collection”, *SAGE Open* 7, 215824401771277. https://doi.org/10.1177/2158244017712774 (2017).

Clickworkers 2018 CLICKWORKERS: Home. URL http://www.nasaclickworkers.com/ (accessed 9.18.18).

Crockett 2016 Crockett, D. “Direct visualization techniques for the analysis of image data: the slice histogram and the growing entourage plot”, *International Journal for Digital Art History* 2. https://doi.org/10.11588/dah.2016.2.33529 (2016).

Dawid and Skene 1979 Dawid, A. P., Skene, A. M. “Maximum likelihood estimation of observer error-rates using the EM algorithm”, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28(1): 20-28. doi: 10.2307/2346806 (1979).

Douglass et al. 2011 Douglass, J., Huber, W., Manovich, M. “Understanding scanlation: how to read one million fan-translated manga pages”, *Image & Narrative* 12: 190–227 (2011).

Downs et al. 2010 Downs, J. S., Holbrook, M. B., Sheng, S., Cranor, L. F. “Are your participants gaming the system?: screening mechanical turk workers”, *CHI 2010 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia, April 2010: 2399–2402. https://doi.org/10.1145/1753326.1753688 (2010).

Duhaime 2018 Duhaime, D. PixPlot. Yale Digital Humanities Lab (2018).

Ekhman and Friesen 1986 Ekman, P., Friesen, W. V. “A new pan-cultural facial expression of emotion”, *Motivation and Emotion* 10: 159–168. https://doi.org/10.1007/BF00992253 (1986).

Han et al. 2014 Han, H., Jain, A. K. “Age, gender and race estimation from unconstrained face images”, Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, MSU Tech. Rep.(MSU-CSE-14-5). (2014).

Han et al. 2015 Han, H., Otto, C., Liu, X., Jain, A. K. “Demographic estimation from face images: Human vs. machine performance” *IEEE Transactions on Pattern Analysis & Machine Intelligence*: 1148–1161. (2015).

Hipp et al. 2013 Hipp, J. A., Adlakha, D., Gernes, R., Kargol, A., Pless, R. “Do you see what I see: crowdsource annotation of captured scenes” *SenseCam 2013 Proceedings the 4th International SenseCam & Pervasive Imaging Conference*, San Diego, California, November 2013: 24–25. https://doi.org/10.1145/2526667.2526671 (2013).

Hipp et al. 2015 Hipp, J. A., Manteiga, A., Burgess, A., Stylianou, A., Pless, R. “Cameras and crowds in transportation

tracking” *WH 2015 Proceedings of the conference on Wireless Health*, Bethesda, Maryland October 2015: 1–8. https://doi.org/10.1145/2811780.2811941 (2015).

Hochman et al. 2016 Hochman, N., Manovich, L., Chow, J. *Phototrails: Visualizing 2.3 M Instagram photos from 13 global cities*. URL http://lab.culturalanalytics.info/2016/04/phototrails-visualizing-23-m-instagram.html (accessed 12.30.16).

Irani 2015 Irani, L. “The cultural work of microwork”, New Media & Society 17: 720–739. https://doi.org/10.1177/1461444813511926 (2015).

James 2014 James, J. Data Never Sleeps 2.0 | Domo. URL https://www.domo.com/blog/data-never-sleeps-2-0/ (accessed 9.20.18) (2014).

Jofre et al. 2018 Jofre, A., Berardi, V., and Brennan, K.. “Time magazine archive: Annotating Faces, Visualizations, and Alternative Applications” (Workshop), *IPAM Culture Analytics Reunion Conference II*, Lake Arrowhead, California, December. (2018).

Jofre et al. 2020a Jofre, A., Berardi, V., Bennett, C., Reale, M., Cole, J.. “Dataset: Faces extracted from *Time* Magazine 1923-2014”, *Journal of Cultural Analytics*. March 16, 2020, https://doi.org/10.22148/001c.12265 (2020)

Jofre et al. 2020b Jofre, A., Cole, J., Berardi, V., Bennett, C., Reale, M. “What’s in a Face? Gender representation of faces in *Time*, 1940s-1990s”, *Journal of Cultural Analytics*. March 16, 2020 https://doi.org/10.22148/001c.12266 (2020)

King and Leonard 2016 King, L., Leonard, P. *Robots Reading Vogue*. Yale DHLab. URL http://dh.library.yale.edu/projects/vogue/ (accessed 11.8.16).

Kittur et al. 2008 Kittur, A., Chi, E. H., Suh, B. “Crowdsourcing user studies with Mechanical Turk”. In: *CHI 2008 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy, April 2008: 453-456. https://doi.org/10.1145/1357054.1357127 (2008).

Kuang et al. 2015 Kuang, J., Argo, L., Stoddard, G., Bray, B. E., Zeng-Treitler, Q. “Assessing Pictograph Recognition: A Comparison of Crowdsourcing and Traditional Survey Approaches”. *J Med Internet Res* 17. https://doi.org/10.2196/jmir.4582 (2015).

LeCun et al. 2015 LeCun, Y., Bengio, Y., Hinton, G. “Deep learning”, *Nature* 521: 436–444. https://doi.org/10.1038/nature14539 (2015).

Leonard and Duhaime 2018 Leonard, P., Duhaime, D. *Yale DHLab - Neural Neighbors: Capturing Image Similarity*. Yale DHLab. URL http://dhlab.yale.edu/projects/neural_neighbors.html (accessed 9.14.18).

Lin et al. 2017 Lin, T., Goyal, P, Girshick, R., He, K., & Dollar, P. “Focal Loss for Dense Object Detection”, *The IEEE International Conference on Computer Vision (ICCV)*, October. https://arxiv.org/abs/1708.02002 (2017).

Lin et al. 2018 Lin, T., Goyal, P, Girshick, R., He, K., & Dollar, P. GitHub Repository. https://github.com/fizyr/keras-retinanet (accessed 2018).

Look Magazine 2012 Look Magazine Photograph Collection, Library of Congress, Prints & Photographs Division. Library of Congress, Washington, D.C. 20540 USA. URL https://www.loc.gov/collections/look-magazine/about-this-collection/ (accessed 9.20.18).

Malli et al. 2018 Malli, R.C., Suri A., & Ramírez S. Github Repository https://github.com/rcmalli/keras-vggface (accessed 2018).

Manovich and Douglass 2009 Manovich, L., and Douglass, J. Timeline: 4535 *Time* magazine covers, 1923-2009. https://www.flickr.com/photos/culturevis/3951496507/ (2009).

Manovich et al. 2016 Manovich, L., Stefaner, M., Yazdani, M., Baur, D., Goddemeyer, D., Tifentale, A., Chow, J. selfiecity, selfiecity. URL http://selfiecity.net/ (accessed 12.30.16).

NYPL Labs 2018 NYPL Labs. The New York Public Library. URL https://www.nypl.org/collections/labs (accessed 9.14.18).

NYPL Map Warper 2018 NYPL Map Warper: Home. URL http://maps.nypl.org/warper (accessed 9.14.18).

Nowak and Rüger 2010 Nowak, S., Rüger, S. “How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation” *MIR 2010 Proceedings of the international conference on Multimedia information retrieval*, Philadelphia, Pennsylvania, March 2010: 557-566. https://doi.org/10.1145/1743384.1743478 (2010).

OpenCV – CiteOpenCV 2017 OpenCV - CiteOpenCV - OpenCV DevZone. URL

http://code.opencv.org/projects/opencv/wiki/CiteOpenCV (accessed 1.5.17).

Organisciak et al. 2012 Organisciak, P., Efron, M., Fenlon, K., Senseney, M. “Evaluating rater quality and rating difficulty in online annotation activities”, *Proceedings of the American Society for Information Science and Technology* 49(1): 1-10. https://doi.org/10.1002/meet.14504901166 (2012).

Parkhi et al. 2015 Parkhi, O. M., Vedaldi, A., Zisserman, A. “Deep Face Recognition” *Proceedings of the British Machine Vision Conference 2015*, Swansea, UK, September 2015: 41.1-41.12. https://doi.org/10.5244/C.29.41 (2015).

Prendergast and Colvin 1986 Prendergast, C., and Colvin, G. *The world of Time Inc.: The intimate history of a changing enterprise*, Volume 3: 1960-1980. New York (1986).

Silberman et al. 2018 Silberman, M. S., Tomlinson, B., LaPlante, R., Ross, J., Irani, L., Zaldivar, A. “Responsible research with crowds: pay crowdworkers at least minimum wage”, *Communications of the ACM* 61: 39–41. https://doi.org/10.1145/3180492 (2018).

Tomnod 2018 Tomnod, Tomnod. URL https://www.tomnod.com (accessed 9.17.18).

Wang et al. 2011 Wang, J., Ipeirotis, P.G., Provost, F. “Managing Crowdsourcing Workers” *The 2011 Winter Conference on Business Intelligence*, Salt Lake City, Utah: 10-12. (2011).

Welinder and Perona 2010 Welinder, P., Perona, P. “Online crowdsourcing: Rating annotators and obtaining cost-effective labels” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops*, San Francisco, California, June 2010: 25–32. https://doi.org/10.1109/CVPRW.2010.5543189 (2010).

Whitehill et al. 2009 Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J. “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise”, *Advances in Neural Information Processing Systems* 22: 2035-2043. (2009).

Yu et al. 2013 Yu, B., Willis, M., Sun, P., Wang, J. “Crowdsourcing Participatory Evaluation of Medical Pictograms Using Amazon Mechanical Turk”, *J Med Internet Res* 15. https://doi.org/10.2196/jmir.2513 (2013).

de Souza 2014 de Souza, R.C. “Chapter 2.3 dimensions of variation in *TIME* magazine.” In T. Berber Sardinha, and M. Veirano Pinto (eds), *Multi-Dimensional Analysis, 25 Years on: A Tribute to Douglas Bieber*, Amsterdam: 177-194. https://doi.org/10.1075/scl.60.06sou (2014).