CrowdMask: Using Crowds to Preserve Privacy in Crowd-Powered Systems via Progressive Filtering

Harmanpreet Kaur[1], Mitchell Gordon[2], Yiwei Yang[1], Jeffrey P. Bigham[4], Jaime Teevan[3], Ece Kamar[3], and Walter S. Lasecki[1]

[1]University of Michigan, Ann Arbor, MI. {harmank,yanyiwei,wlasecki}@umich.edu
[2]University of Rochester, Rochester, NY. m.gordon@rochester.edu
[3]Microsoft Research, Redmond, WA. {teevan,eckamar}@microsoft.com
[4]Carnegie Mellon University, Pittsburgh, PA. jbigham@cs.cmu.edu

Abstract

Crowd-powered systems leverage human intelligence to go beyond the capabilities of automated systems, but also introduce privacy and security concerns because unknown people must view the data that the system processes. While automated approaches cannot robustly filter private information from these datasets, people have the ability to do so if the risk from them viewing the data can be mitigated. We present a crowd-powered approach to masking private content in data by segmenting and distributing smaller segments to crowd workers so that individual workers can identify potentially private content without being able to fully view it themselves. We introduce a novel pyramid workflow for segmentation that uses segments at multiple levels of granularity to overcome problems with fixed-sized approaches. We implement our approach in CrowdMask, a system that allows images with potentially sensitive content to be masked by appearing in progressively larger, more identifiable segments, and masking portions of the image as soon as a risk is identified. Our experiments with 4134 Mechanical Turk workers show that CrowdMask can effectively mask private content from images without revealing sensitive content to constituent workers, while still enabling future systems to use the filtered result.

Introduction

An increasing number of crowd-powered systems require workers to interact with user-generated data, such as *audio recordings* (Lasecki et al. 2012), *personal photographs* (Bigham et al. 2010; Merritt et al. 2017), *email* (Kokkalis et al. 2013), *documents* (Bernstein et al. 2010), *search queries* (Bernstein et al. 2012), program code (Chen et al. 2017) and *handwritten text* (Little and Sun 2011; Chen et al. 2012). These systems can accidentally expose information that users would like to remain private to the workers powering the system, because similar information is required to complete the task. For instance, a blind user of VizWiz (Bigham et al. 2010) may want the crowd to identify the name of a prescription medicine from a photograph of the bottle. Because prescription labels typically contain the patient's name, the crowd can only provide an answer without learning the user's identity if the patient's name is obscured, but the medicine's is not.

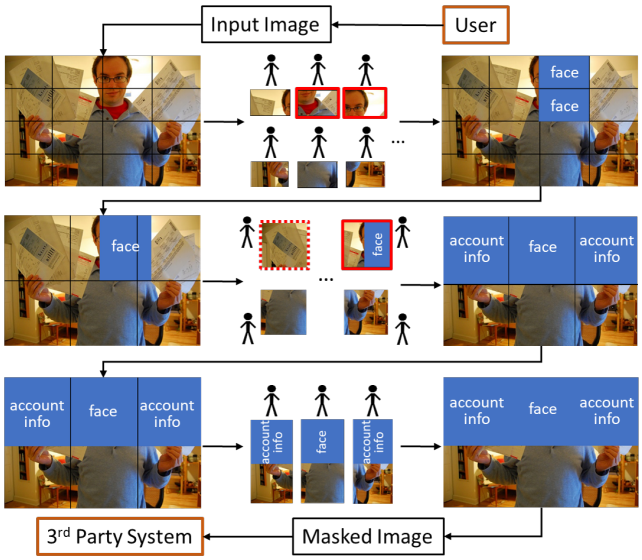

Figure 1: Our crowed-powered content filtering pipeline. Before users submit content that may contain sensitive data to a third-party crowd-powered system, our multi-level filtering approach progressively shows larger segments of it to successive workers, filtering out potentially sensitive content at each step. Sensitive data is hidden while minimizing the risk that any one worker sees too much. The resulting (masked) version of the content is then forwarded on to the intended third-party task.

If sensitive content could be easily and robustly filtered from larger user-generated datasets, such privacy threats could be mitigated. Unfortunately, even the best automatic approaches can fail because they require a rich understanding of the content and the ways that it might be used. For example, an automated system may help a user filter their account number from a picture of their bank statement by masking all the numbers in the picture, but this approach would make it impossible to then extract the customer service phone number from the picture.

In this paper, we introduce a pre-processing step in the crowd pipeline: a crowd-powered system that uses human intelligence to mask private information in user-generated data (Figure 1). Using our system, people can provide natural language descriptions of what to filter (e.g. "hide any

embarrassing content"), which requires no technical skills or knowledge of the system's underlying processes. To avoid any one worker from gaining access to potentially private information, we introduce a pyramid workflow for dividing content for filtering, showing each worker only a small portion of the original content. Our workflow employs multiple segmentations at different levels of granularity to avoid problems arising from unknown content granularity (e.g. the system does not know a priori how large a face is in a given image). The segment granularity is optimized by the system based on user-specified budget constraints.

We evaluate our system and approach using images, a common type of user-generated data for crowd-powered systems (e.g. (Bigham et al. 2010)). Our experiments with 4134 Mechanical Turk workers show that we can mask a variety of private information from images, while making it possible for a separate crowd of workers to perform the originally-requested task. We make the following contributions:

- Identify relevant factors for building a crowd-powered system that can mask private content in user-generated data (e.g., images).
- Introduce a pyramid workflow for multi-level content segmentation which ensures that no individual worker can access all potentially private information.
- Build a crowd-powered system for filtering private content from images based on natural language queries.

Related Work

We begin by outlining the types of tasks that crowdsourcing platforms employ that require workers to interact with end-user information, and discuss the threats crowd workers pose to such systems.

Crowd-Powered Systems May Expose User Data

Many crowd-powered systems assist users in their daily lives, often using data from users. For example, Soylent helps users edit documents (Bernstein et al. 2010), and PlateMate determines how many calories meals contain based on photographs of them (Noronha et al. 2011). The intended tasks are generally not expected to contain sensitive information, but nevertheless may. For example, PlateMate may receive an image from a diner at a restaurant that accidentally includes a credit card on a table. Because crowd-powered systems can easily be confused with automated systems by end users, exposure can happen unintentionally. Interactive crowd-systems that respond to users in real time (Lasecki et al. 2011) also make it easy to mistakenly capture sensitive information. Getting responses from the crowd in a few seconds (Bernstein et al. 2011) means there is little time for users to review the content they are sending.

Assistive technologies are a natural match with crowdsourcing because they provide mediated access to human assistance. Scribe (Lasecki et al. 2012), for example, provides deaf and hard of hearing users with real-time captions, and VizWiz (Bigham et al. 2010) allows blind users to get answers to visual questions. These systems can have a profound impact. VizWiz has answered over 80,000 questions for thousands of users. However, users of these systems may be unable to effectively avoid capturing sensitive information. For example, a blind user might not be able to tell that they have inadvertently captured a billing statement in an image sent to VizWiz (Ahmed et al. 2016), and a deaf user might not be able to tell that their account information could be overheard in speech until after it has been captioned by Scribe (Lasecki et al. 2012).

Although not entirely anonymous, requesters tend not to know the identity of crowd workers on platforms like Amazon Mechanical Turk (Lease et al. 2013). This relative anonymity, coupled with a range of worker skill levels and the need for workers to complete large numbers of tasks to earn a reasonable wage, creates the need for quality control systems (Bernstein et al. 2010; Ipeirotis, Provost, and Wang 2010). These approaches increase the overall quality of the work, but at a cost; they tend to increase the number of workers who will see each piece of information contained in a task. Crowd-powered systems that use personal information potentially put users at risk of identity theft, blackmail, and other information-based attacks.

Crowd-Based Privacy and Security Threats

Concerns with issues related to the privacy and security of sensitive information used in crowd-powered systems have led to some initial work exploring the types of problems that may arise. Harris et al. (Harris 2011) bring up the idea that ordinary workers might be hired for potentially malicious tasks. Lasecki et al. (Lasecki, Teevan, and Kamar 2014) outline a variety of different individual and group (both coordinated and uncoordinated) attacks that are possible on current platforms, and demonstrate that workers can be hired to do seemingly-malicious tasks (such as copy a credit card number from another task), even if some percentage of workers will abstain from such tasks. Teodoro et al. (Teodoro et al. 2014) also found similar hesitation to potentially illicit tasks, such as mailing lost cell phones to a service promising to find their owner and return them. Forums and other worker communities also help discourage this behavior. Our experiments investigate protecting image data of the kind dealt with by VizWiz when answering visual questions for blind users, where information from bank accounts, to names and addresses, to accidentally revealing images (e.g. accidentally capturing unintended information) may arise. Malicious workers may begin targeting such systems as their popularity grows, the information captured becomes more valuable, and these incidents become more frequent (Lasecki, Teevan, and Kamar 2014).

Approaches to Preserving Privacy

To preserve privacy in crowd systems, Wang et al. (Wang et al. 2013) studied how to detect malicious workers. Most other approaches have focused on protecting private or sensitive content itself. Varshney (Varshney 2012) proposed using visual noise and task separation to preserve the privacy of content in images. This could help protect some types of information, but in many cases information needed to complete the final task (e.g., read a label for a blind user) is lost. Little and Sun (Little and Sun 2011) looked at protecting privacy in medical records by asking workers to first annotate a

blank record to indicate where field values are entered, then using this information to divide a real medical record into pieces that workers could help transcribe without being able to see too much information. Swaminathan et al. (Swaminathan et al. 2017) recently introduced WearMail, a system that searched for content in a user's private emails safely by using crowds to provide examples of a known pattern to look for within emails, without ever seeing the content of any emails.

We attempt to counter these threats by ensuring that no worker individually is able to see enough information to do the end user harm. It differs from existing approaches in that: i) the division algorithm progressively zooms out while applying partial masks along the way, overcoming many of the context-based challenges encountered in previous work (e.g. information referenced in other pieces), ii) it uses a general model which does not require an initial template that is used to advise the division of future tasks, and iii) the final task being completed does not need to be known a priori. While few systems can prevent coordinated groups from attaining information from tasks, protecting against individual worker threats drastically decreases the threat to end users. To our knowledge, ours is the first work to explore such approaches to general, task-independent privacy preservation using an implemented system.

Preliminary Studies

Our approach for filtering private content relies on dividing the original content into smaller segments, and masking the potentially private content in these segments. We conduct three preliminary studies to test: (1) if accidentally sharing private information is a significant problem, (2) if our proposed approach that relies on dividing and masking content is feasible, and (3) the ideal approach for instructing workers to perform the masking task.

Study 1: Significance of the Problem

We used the original VizWiz dataset (filtered version available at **`http://vizwiz.org/data/`**) to get of sense of the privacy threats present in user-generated data created in crowd-powered systems. Images from VizWiz were taken by blind users of the application (Bigham et al. 2010), and often contained personally identifiable information (PII) that the user chose to include (e.g. a credit card number they wanted to know) or that was accidentally included because the user could not see PII included in the background (*e.g.*, a person's face behind an object of interest). Out of a total of $47,005$ images in the pre-release VizWiz dataset, 7.37% (or $3,462$) images contained PII (*e.g.*, face, name, address, ID number). The images with PII were manually filtered from the dataset because automatic approaches cannot robustly filter all PII. We sampled 200 of the images with PII with an automated face detector. 126 out of 200 sampled images contained a face, but only 47 (37.30%) of these 200 were flagged by a state-of-the-art automated face detector[1]. Private information in user-generated data remains a privacy threat as long as systems use only automated approaches to filter it.

[1]https://aws.amazon.com/rekognition/

Study 2: Feasibility of the Content Division Approach

As an initial test of the feasibility of our approach, we explored the segmentation level at which people can determine what class of object they are viewing, *e.g.*, an arm, a face, a keyboard, versus the level at which they can identify a specific instance of an object, e.g. the identity of a person. We used images of people and objects that had been cropped to the size of the entire image (400px by 400px) so that the relative size of the object in the image would not be a factor. The following experiments used Mechanical Turk workers, each of whom were paid \$0.14-\$0.16 per task.

Class Recognition To evaluate whether the crowd can recognize objects at increasing levels of granularity, we showed 60 Mechanical Turk workers two images—a face and a credit card—at three levels of granularity, and asked them to identify what type of object they saw using a multiple choice question with five plausible answers. Each level of granularity received responses from 10 unique workers, and no worker could answer for more than one level of granularity of a given image.

When shown half of a credit card, 100% of workers were able to correctly identify it as a credit card. When shown $\frac{1}{5}$ and $\frac{1}{10}$ of a credit card, 80% and 70% of workers correctly identified it, respectively. However, when trying to recognize a face, the worker success rate dropped far more quickly. When showing $\frac{1}{2}$ and $\frac{1}{5}$ of a face, 100% of workers correctly identified that they had seen a face. However, at a granularity level of $\frac{1}{10}$, that number dropped to 40%.

Identity Recognition We evaluated the effectiveness of the division approach in hindering workers' ability to identify a person given different sized image segments. To do so, we showed 125 Mechanical Turk workers 25 segments of a face: 16, 6, and 3 segments at three levels of granularity. Each segment was viewed by 5 workers, and we asked them the question "Does this image contain a face?" Then, before ending the task, we presented workers with a police-lineup style interface showing six images of faces side-by-side in random order. One of these images was a different picture of the same person they saw in the previous screen, but in a slightly different setting. We avoided using the same image of the person to avoid other pieces of the scene being used to identify the matching image. The other five images in the lineup were all people of the same race, gender, and approximate age, but were identifiable as different people.

We found that, when simply showing workers a full image of the face without any level of granularity, workers correctly recognized that person around 60% of the time. When segmenting the face into just three segments, the rate of recognition dropped to just 13%. When segmenting the face at a much higher level of granularity (i.e., 16 segments), the recognition rate dropped even further to 7.9%.

Conclusion The key finding from these initial studies is that workers can identify the *class* of an element in an image even when it is divided in half (100% of workers got the answer correct), whereas only 13% were able to determine the *identity* of the person in the photo. In naive image

segmentation (where a single division level is used), the segmentation must divide content by luck to a sufficiently small size such that the identity remains unknown. Decreasing the size of segments will increase the chances of dividing arbitrary content into pieces too small to identify private content, but it also increases the chance that some element is divided beyond workers' ability to accurately identify its class. This suggests that the use of a progressive, multi-level approach will allow the system to filter what can be identified with smaller pieces of information, e.g. a generic class like face or credit credit, while not revealing too much. Such filtering can be done by masking the smaller piece of the image identified as potentially private. Then, when the masked segment is shown in a larger segment (at a lower level of granularity), entity or identity recognition is not possible.

Study 3: Framing the Question for Crowdworkers

Another important consideration when relying on crowdworkers to mask private content is how this filtering task should be framed. We tested four variants of instructions for this purpose. Each instruction type was evaluated using the three images: (1) an assortment of cards lying on a table, two of which are credit cards; (2) a man holding pieces of paper, some of which were bills with balances and other information (no account information visible); and (3) a fully addressed and stamped letter. We segmented each image into 25 segments total (16, 6, and 3 segments at different levels of granularity); each segment was seen by three Mechanical Turk workers. We tested two different phrases for private content: "PII" and "sensitive information" (1821 total workers over four question types, paid $0.14 - $0.16 per task).

- **(F1) Filter Question:** Workers are shown only the filter definition. For example, "Does this image contain any potentially sensitive information about the requester?"
- **(F2) Filter Question with Example:** The initial question, with examples of the type of information that should be filtered. For example: "If the image contains a face, name, address, or other contact information, click 'yes'."
- **(F3) Filter Question with Example and Non-Example:** The initial question, with non-examples that give workers an idea of the type of information that should *not* be filtered: "If it contains a company name or non-identifying documents, click 'no'."
- **(F4) Filter Question with Non-Examples and End Goal:** The question, examples, non-examples, and the task's goal that tells workers what information we ultimately need from the image after filtering. The hope is that this information will prevent workers from electing to mask segments that make the goal (e.g. answering "Which of these is my library card?") impossible.

We observed that worker responses varied significantly by each instruction type. F1 was subject to each worker's idea of what the question meant. This led to both false-positives and false-negatives because of high disagreement among workers: they often correctly identified an object that they saw, but disagreed on whether that object needed to be filtered. F2 gave the workers an idea of what the question was looking for, but resulted in a larger number of false-positives. In 5 of the 6 runs, the crowd masked more than half of the image, a ratio that is significantly higher than the results from our study 2 and higher than results obtained with any other instruction type. F3 attempted to rein in the false-positives, but we found that workers often did not listen to our non-example. For instance, the crowd masked a face even when specifically asked not to. Finally, F4 produced images closest to our baseline, with unwanted information masked, but enough information left to answer the end-goal. As a result of this preliminary experiment, our system used the instructions in F4.

CrowdMask

CrowdMask is a crowd-powered system that can filter arbitrarily-sized private information from user-generated data (Figure 1). It protects end users from sharing potentially private content on other crowd-powered systems (such as VizWiz) by masking such content before the original post is sent to the crowd. Users define "filters" in natural language for data sent to the crowd. Instead of having workers try to complete the original task from a single smaller segment of the image, our approach acts as a filtering step preceding a crowd task. This makes the (often very necessary) context in the larger scene available to the workers completing the primary task.

Basic Approach: Dividing Content

Our basic approach is to divide content into pieces that each contain incomplete information. Based on the results of our preliminary study, we know that this division approach is feasible in recognizing potentially private content without disclosing the identity of an individual to any crowd-worker. This can greatly reduce the risks faced by end users, but is sensitive to the type of risk, granularity, and information available in specific instances. For instance, an image may contain multiple types of PII: a person's face, their name on a nametag, and a partial reflection of them in a mirror. Each of these sources of information can be a different size, in a different location, and may be identifiable in different ways.

The challenge with this basic approach is setting an appropriate granularity for the segmentation. A single level of granularity might allow the pieces of PII to be separated from one another, but each piece might still be contained in a segment. Setting the granularity higher might result in the person's name tag being filtered out successfully without anyone seeing their full name or job title, but also might result in the person's face being divided into pieces too small to identify that each one is part of a face — resulting in no piece of the face being filtered, so the user's face remains unmasked in the final image.

Our Approach: The Pyramid Workflow for Minimal-Knowledge Filtering

To solve the granularity identification problem, we use a *pyramid workflow* that first presents very small segments of the image, and then progressively zooms out to identify visual information at different scales. Workers are first shown

the smallest possible segments, which is least likely to reveal sensitive information. Once all potentially private segments of a size have been masked, new workers are shown larger segments with prior masks applied to the images (Figure 1). This process continues until workers have masked the image at all granularities.

Consider the previous example of an image that contains both a face and a nametag. Applying multi-level filtering allows both the nametag and face to successfully be masked without any worker seeing either. Initially, workers see only small segments of the image, which allows them to mask the nametag. It is difficult or impossible to tell that such small segments contain faces. Filtering faces requires subsequent, larger segments: these larger segments now have the nametag masked by the filtering that took place at the smaller size. This allows workers to identify all regions that should be masked without revealing sensitive information.

Optimizing Segment Sizes

As mentioned before, content can vary greatly in size and type of information. To use our multi-level approach effectively, there must be a difference between each level so that workers can gain new context and recognize potential threats that they could not in the previous level.

To do this, we optimize the separation in segment sizes between different zoom levels, given a cost bound. We start with the maximum amount that a user is willing to pay to filter each query. Given the user's budget B, and the cost of a crowd task C, we can compute the total number of questions N that we can have answered, $N = B/C$.

Using this bound, we then select a number of levels L to use. The selection of this value is dependent on the type, size, and quality of content that will be used. For instance, high resolution images may require more levels to effectively filter because both small distant objects and large close objects may contain legible information. Alternatively, given a [minimum] growth rate G_m, we can optimize the number of possible levels subject to the budget by adding levels that contain G_m times as many segments as the previous level until the total number of segments no longer fits within the specified budget. We then optimize the number of segments to maximize separation by either solving for a precise answer or using a numerical approximation if no exact answer is possible. In preliminary trials we found that 3 levels effectively handled content seen in web/phone images. Assuming $L = 3$ for our example, we can set up a linear system to find the segment sizes for each level that maximizes separation:

$$N = \sum_{i \in 0 \ldots L} N_i = N_1 + N_2 + N_3$$

Where N_i is the number of segments to be created at level i. Now, we want to find the growth factor G between levels. Redefining this linear system as a function of G gives:

$$N = N_1 + G*N_1 + G*(G*N_1)) = N_1 + G(N_1) + G^2(N_1)$$

We set a minimum division size for N_1 of M (where $M = 2 * 2 = 4$ is the smallest non-trivial segmentation). This is used as the division for the smallest size of the image at the lowest zoom level, while the number of segments along the other dimension of the image is calculated proportionally to the aspect ratio of the image (e.g., a 2:1 aspect ratio image would end up divided into $2 \times 4 = 8$ segments). This gives:

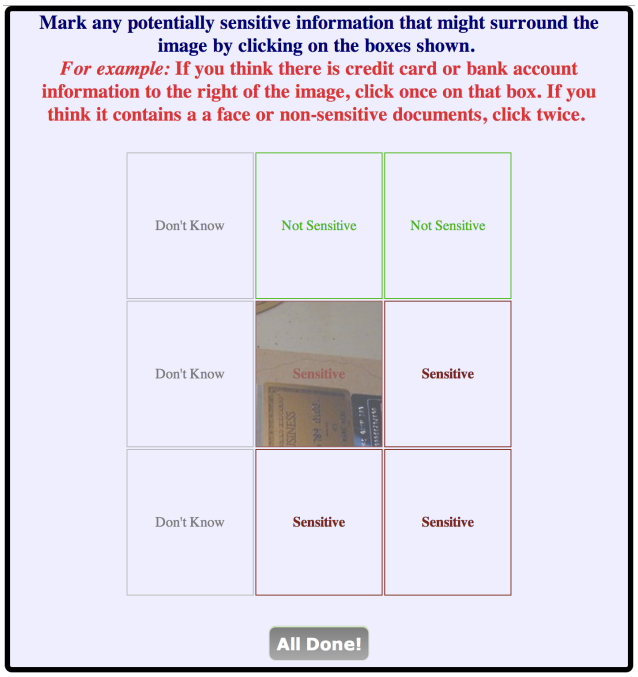

Figure 2: The content prediction UI allows workers to label likely-sensitive adjacent segments without seeing them.

$$N_1 >= M \text{ so } N_1 + G(N_1) + G^2(N_1) = M(1 + G + G^2)$$

Now we can factor this term to find the solutions for G. Note that while this case can be solved using the quadratic formula, not all selections of L will lead to such clean forms — for instance, $L = 6$ results in a fifth degree polynomial that cannot be factored. In these cases, there are numerical methods can find solutions well within a reasonable margin of error. Finally, we get our growth rate:

$$G = \sqrt{N/M - 3/4} - 1/2$$

We use this algorithm to generate segments with maximal separation, which is ideal for our approach. Thus, users need only define a price they are willing to pay any time before sending their source image (e.g., via a system setting).

Predicting Sensitive Content

In addition to directly showing workers images, we can also leverage their understanding of the scene to predict what is in adjacent segments without showing them. For instance, if we are filtering for PII and workers observe someone's body in one image, then it can be reasonably assumed that their face may be visible directly above it. The system interface thus includes a second stage (Figure 2) that asks workers to indicate whether the segments surrounding the original segment are: *(i)* very likely to contain sensitive information, *(ii)* very likely *not* to contain sensitive information, or *(iii)* they are unsure of what would be contained (the default answer).

To best use workers' ability to predict sensitive content outside their current view, we issue images in each level of our process in two interleaved "checkerboard" patterns, where each segment shown in the first pass is bordered directly above, below, left, and right by content that has not yet been seen by another worker. After this first pass has been

completed, images from the alternate segments are issued to arriving workers. Doing this allows us to withhold asking about segments from the first pass with sufficient agreement between workers on content being or not being present. In the best case, over 50% of the segments from a level can be answered without the need for workers to even view the segment itself (some content can be filtered from the diagonal elements in the first level). While we expect such extreme cases to be rare, there is the possibility for large privacy gains and cost savings.

System Components

CrowdMask is comprised of three main components: an end user session creation process, a front-end interface for crowd-workers to complete their specified task, and server-side image modification framework.

The end user session creation process allows the user to specify an image to be filtered, the number of granularity levels to use, the maximum amount they would like their session to cost, and the instructions that are shown to workers. Their maximum cost determines what size segments each level of granularity will use.

The worker interface shows an image and a "yes" or "no" question, such as "Does this image contain any sensitive content?" Filtering is done on a per-segment basis to simplify the task and make consensus-finding more tractable. Finally, workers are asked to predict whether there may be any sensitive content that surrounds the image, using a $3x3$ grid that contains the original segment, surrounded by empty, clickable boxes that allow workers to mark the predicted content of each box as either "sensitive," "not sensitive," or "don't know." The system serves workers segments starting with the highest zoom level. Importantly, images are segmented and filtered server-side to prevent workers from being able to bypass our restrictions.

Semantic Labeling Workers — both those helping to filter content, as well as those who contribute to the final task that our masked image is input to — need some context to correctly complete their task. While our masking process filters much of the context that may create a privacy threat for the requester, this context can be partially obtained using the descriptions of the content being hidden. To provide this, we collect a 1-2 word label from workers which describes the original segment they were reviewing. This label can then be applied to the mask that covers the content in subsequent levels. While this information is only useful to future workers in the case when the original content is masked, we always collect a label to prevent this additional effort from biasing workers towards labeling as not sensitive.

Evaluation of the Approach

We evaluate our approach based on how well the crowd-workers are able to mask and predict potentially private information in images. We compare their results to ground truth data generated by two coders for each image used for evaluation. Each coder marked the private content for all images, and ground truth was computed based on an agreement between them. Inter-rater reliability was calculated using Cohen's kappa, with a score of 0.72. We measure precision, recall and F1 scores per segment by comparing workers' answers to ground truth data to calculate true positives, false positives, true negatives and false negatives.

Identifying and Masking Private Content

To evaluate how well workers can filter within the multi-level zooming and masking model, we ran an experiment that involved filtering for both sensitive information and personally identifiable information. Four images were tested for each of these conditions. The first image contained an assortment of cards lying on a table, two of which were credit cards. The second contained a man holding pieces of paper, some of which were bills with balances and other information on them (no account information was visible to workers). The third contained a fully addressed and stamped letter. The fourth image contained a police protest/arrest scene in which two faces of protesters were visible. We used three levels of granularity when filtering the images. The highest level was comprised of 16 segments, the middle level 6, and the lowest 3 (total 25 segments).

For each of the four images, we ran three trials with "sensitive information" and three with "personally identifiable information (PII)" filtering instructions. Each image was divided into 25 segments across three levels of granularity, and each segment was shown to three workers (total 1702 responses after filtering). Averaging the scores from total 24 runs (4 images x 6 trials/image), we found that at a threshold of 50% agreement, the end result images had a precision of 38.2% and recall of 94.8%. When raising the agreement threshold to 80%, there is an increase in precision to 52.9% and a decrease in recall to 66.5%. The magnitude of this change suggests that there is a high level of agreement between workers. Figure 3 shows the full span of precision and recall scores for the final, masked image that is generated. While precision is low (in part because we filter by segment), our high recall means that sensitive information is rarely left unfiltered.

Comparison to Single-Level Segmentation Figure 4 shows precision and recall scores when just considering the first (highest) level of granularity. At a threshold of 50% agreement, the single-phase filter had a precision of 46.2% and recall of 62.0%; at a 90%, precision is 54.6% and recall is 36.5%. Compared to the multi-level run, this resulted in a significantly lower recall but slightly higher precision. We also compared the F1 score (which aggregates precision and recall) for all four images of single-level to multi-level segmentation and found that there was a significant 29.8% improvement between single and multi-level, where F1 increased from .438 to .591 ($p < .01$).

Predicting Private Content

Along with identifying and masking private content, we also wanted to know if workers could effectively predict what is in segments adjacent to the image segment they were shown, without needing to actually show workers the content of those segments (similar to the handwriting completion by prediction in (Zhang, Lai, and Bcher 2012)). To evaluate

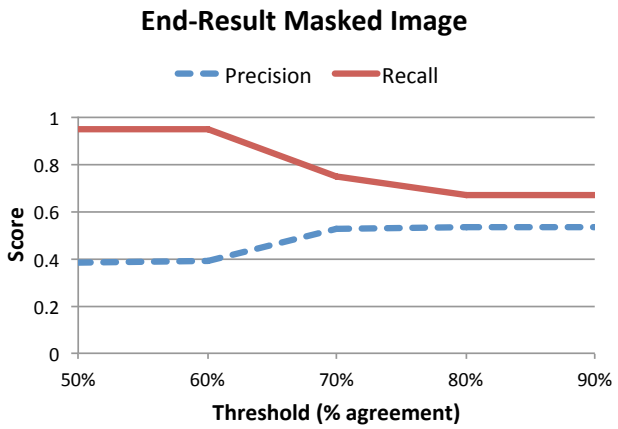

Figure 3: Average precision and recall for 6 runs on 4 images. We found that at 50% agreement, the resulting precision is 38.2% and recall is 94.8%. When the agreement is 80%, there is an increase in precision to 52.9% and a decrease in recall to 66.5%.

this, we chose a test set of image segments that fit under five categories: *(i)* information partially in scene, *(ii)* information out-of-scene with clear indicator, *(iii)* information out-of-scene with expected content, *(iv)* information out-of-scene with expected non-content, and *(v)* information out-of-scene with unexpected content / no-content baseline. We selected two images in each category (total: 10 images), and got 157 worker responses, with each image shown to 15 workers.

This prediction process can be used to show fewer segments to workers for masking: we calculate the total number of predicted "Yes"s, "No"s and "Maybe"s per segment, and select the top-voted element (or multiple elements, given a tie) for masking, as long as the number of "Yes"s was above a threshold of 50%. Doing this gave an average precision across all images of 94.8%, with an average recall of 92.5%.

Detecting Embarrassing Content

To evaluate the effectiveness of our approach at understanding highly subjective scenarios, we used three potentially embarrassing images, which contained scenes, such as a person picking their nose. We used three levels of granularity, and asked, "Does this image contain anything embarrassing?" Two images contained one segment with embarrassing content, while the third image did not. We received responses from 239 workers with each segment viewed by 3 workers, and observed that the crowd was able to identify both of the potentially embarrassing situations with a perfect score for both precision and recall, masking only the segments that contained the embarrassing scene.

Evaluation of the System

Our system is a pre-processor for images sent to crowd-powered systems: ideally, it should mask the private content in images, but still enable crowd-workers to complete the original task posted by the user. We evaluate our system on whether the end goal is answerable using the masked image.

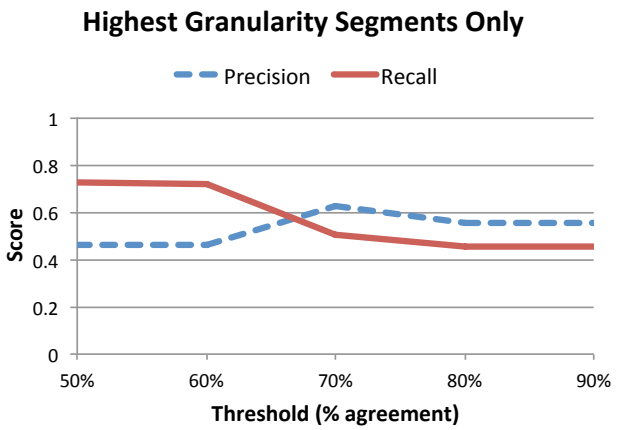

Figure 4: Average precision and recall scores from just the highest level of granularity in the main experiment. Averages are across all tests from all four instruction types.

End-Goal Answerability

Key to the success of our system is the ability for workers to still answer a desired question by looking at an image that has been masked. To evaluate this, we used three images that were filtered from the mutli-level masking experiment. We then presented these masked images to 30 workers, along with the image's associated question: "How many pieces of paper is this person holding?", "Which of these is my loan package checklist?", and, "What is in this picture?" We asked workers whether they believed they would be able to answer the question with the masked image they had been shown, and then to simply answer the question. Each question was shown to 10 unique workers. 90% of workers replied that they thought they could answer the question, and 90% of workers answered the question correctly (though the two sets are not a necessarily a union). This demonstrates that content filtering, even using the somewhat course-grained (medium-low budget) divisions we did in our trials, can be done in a way that still allows for the end goal question to be answered.

Reducing the Cost of Masking

The cost of running our system is largely dependent on the levels of granularity that the user wishes to filter at, the size of the image, and the level of worker redundancy desired. We priced our tasks at an average of $0.15, and used around 25 segments per image in total, across all levels. While this means each image would cost $3.25 to filter, we specifically did not optimize for price, but instead focused on showing that it is possible to protect user information using a natural language-defined filter. For comparison, reducing the pay rate of these tasks to the U.S. minimum wage ($7.25), would result in images costing $1 each to filter.

Reducing Task Size By using the crowd's predictive capabilities, we can reliably reduce the number of segments that need to be individually marked by workers by 17% while retaining 90% accuracy (Figure 5). This reduces the number of tasks that workers need complete. The number

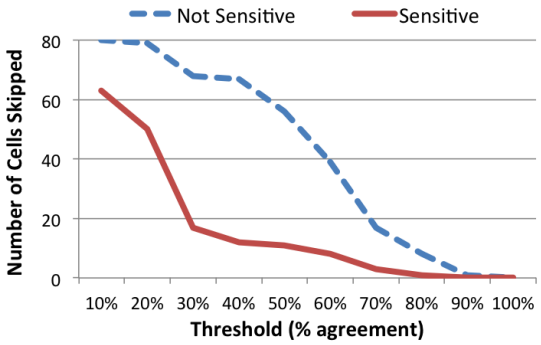

Figure 5: Plot of how many tasks would be skipped based on worker prediction in our example scenario as agreement rate threshold is increased.

of granularity levels and max-cost can also be optimized for images of certain sizes, quality levels, and content-types. Increasing the number of images shown to workers per task would allow the filter cost per segment to be decreased because workers can more easily complete sets of tasks. To ensure that this does not undermine our core approach, the image segments should be drawn from disjoint pieces of the image, and comprise only a relatively small percentage of the image per worker.

Automated systems can also be used as a 'first pass', filtering content of a type that might present a risk. Computer vision approaches cannot accurately distinguish the context in which certain types of information are sensitive, but they can guide the human-filtering process by having workers look only at segments that contain potentially harmful *types* of information. This can even guide the use of CrowdMask at a higher level. For instance, an image not containing numbers is unlikely to contain harmful account information, so no filtering must be done by human workers.

Limitations

Our results demonstrate that we can accurately and reliably filter sensitive content from images based on natural language definition – even with subjective queries – without revealing the information to workers along the way. We show that this content masking is done in such a way that the initial question behind the image can still be answered.

To fully understand how CrowdMask would be used in practice requires recruiting people to create filters for their own potential tasks. However, many tasks that would benefit most from our approach are not currently being run due to the risk of exposing private information. Running end-to-end experiments on an untested system poses substantial risk for the end user. For this reason we focus on understanding our system and its capabilities. By learning how workers completed their task and establishing the capabilities of the system, we hope to enable crowdsourcing researchers and system builders to mitigate privacy threats in their systems.

While the our system is designed to thwart individual attacks, the information filtered remains at risk from coordinated group attacks (Lasecki, Teevan, and Kamar 2014). If a sufficient number of workers colluded and shared images, they would likely be able to recreate the original content in the images. Although communication channels are available to workers (e.g., forums) (Irani and Silberman 2013), they tend to ally with requesters against coordinated attacks.

Generalizing to Other Content Types

While future work is needed for further development and experimentation, our pyramid workflow can be applied to other forms of media. Below, we discuss how this could be accomplished in future systems and extensions of CrowdMask:

- **Text:** The workflow remains the same, but works on the simpler 1-dimensional case of linear text where segment size is measured in number of words. When workers see words or phrases that might constitute sensitive information, they should be able to click them and filter them out. We are currently building a module that extends CrowdMask for this purpose.
- **Audio:** Segmenting audio can be done in a similar manner to text, but because the word boundaries are not clearly discernible, words may be truncated by segmentation, even with automated segmentation assistance. To solve this, a system can stagger responses to ensure complete coverage. Selecting the length of audio clips has the same trade-offs as image segmentation, larger clips provide more content, leading to more context and more risk.
- **Video:** Like image, video requires a 2D filter, as well as temporal division to avoid over-filtering. If audio is present, that can be handled together, or as a separate task (giving less context, but potentially more secure).
- **Other Media:** Other forms of media, including hypermedia and structured content (e.g., databases or knowledge graphs) can also be handled if the appropriate segmentation methods are added. Structured forms of data may have an advantage in terms of privacy protection and cost because there is more a priori knowledge about constraints and templates for information.

Conclusion

In this paper, we have introduced a generalizable, task-independent approach for filtering potentially sensitive information using the crowd. While automated approaches can only remove content in a very coarse-grain fashion (i.e., based on type), we have shown that it is possible to use human intelligence to filter content based on simple natural language queries, without exposing information to the workers themselves. Our novel pyramid workflow progressively "zooms out" from a fine-grained content segmentation (many small pieces) to a more coarse-grained segmentation (fewer large pieces), masking content as soon as it can be identified as something the user did not want included in their task request. We showed that this approach can effectively filter image content, while hiding sensitive information from constituent workers.

Acknowledgments

We would like to thank all of our study participants for their help, and Meredith Ringel Morris for her feedback on this work. This work was supported in part by Microsoft Research, Google, IBM, and the University of Michigan.

References

Ahmed, T.; Shaffer, P.; Connelly, K.; Crandall, D.; and Kapadia, A. 2016. Addressing physical safety, security, and privacy for people with visual impairments. In *Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS '16)*, 341–354.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2010. Soylent: A word processor with a crowd inside. UIST '10, 313–322.

Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. UIST '11, 33–42.

Bernstein, M.; Teevan, J.; Dumais, S. T.; Libeling, D.; and Horvitz, E. 2012. Direct answers for search queries in the long tail. CHI '12, 237–246.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; and Yeh, T. 2010. Vizwiz: Nearly real-time answers to visual questions. UIST '10, 333–342.

Chen, K.; Kannan, A.; Yano, Y.; Hellerstein, J. M.; and Parikh, T. S. 2012. Shreddr: Pipelined paper digitization for low-resource organizations. DEV '12, 3:1–3:10.

Chen, Y.; Lee, S. W.; Xie, Y.; Yang, Y.; Lasecki, W. S.; and Oney, S. 2017. Codeon: On-demand software development assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 6220–6231. ACM.

Harris, C. G. 2011. Dirty deeds done dirt cheap: A darker side to crowdsourcing. In *Privacy, security, risk and trust (passat)*, 1314–1317. IEEE.

Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *HCOMP Workshop*, 64–67.

Irani, L. C., and Silberman, M. S. 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. CHI '13, 611–620.

Kokkalis, N.; Khn, T.; Pfeiffer, C.; Chornyi, D.; Bernstein, M. S.; and Klemmer, S. R. 2013. Emailvalet: Managing email overload through private, accountable crowdsourcing. CSCW '13.

Lasecki, W. S.; Murray, K. I.; White, S.; Miller, R. C.; and Bigham, J. P. 2011. Real-time crowd control of existing interfaces. UIST '11, 23–32.

Lasecki, W.; Miller, C.; Sadilek, A.; Abumoussa, A.; Borrello, D.; Kushalnagar, R.; and Bigham, J. 2012. Real-time captioning by groups of non-experts. UIST '12, 23–34.

Lasecki, W. S.; Teevan, J.; and Kamar, E. 2014. Information extraction and manipulation threats in crowd-powered systems. CSCW '14, 248–256.

Lease, M.; Hullman, J.; Bigham, J. P.; Bernstein, M.; Kim, J.; Lasecki, W. S.; Bakhshi, S.; Mitra, T.; and Miller, R. 2013. Mechanical turk is not anonymous. *SSRN*.

Little, G., and Sun, Y.-a. 2011. Human ocr: Insights from a complex human computation process.

Merritt, D.; Jones, J.; Ackerman, M. S.; and Lasecki, W. S. 2017. Kurator: Using the crowd to help families with personal curation tasks. In *CSCW*, 1835–1849.

Noronha, J.; Hysen, E.; Zhang, H.; and Gajos, K. Z. 2011. Platemate: Crowdsourcing nutritional analysis from food photographs. UIST '11, 1–12.

Swaminathan, S.; Fok, R.; Chen, F.; Huang, Ting-Hao Kenneth, L. I.; Jadvani, R.; Lasecki, W. S.; and Bigham, J. P. 2017. Wearmail: On-the-go access to information in your email with a privacy-preserving human computation workflow. UIST '17.

Teodoro, R.; Ozturk, P.; Naaman, M.; Mason, W.; and Lindqvist, J. 2014. The motivations and experiences of the on-demand mobile workforce. CSCW '14, 236–247.

Varshney, L. R. 2012. Privacy and reliability in crowdsourcing service delivery. In *SRII 2012*, 55–60. IEEE.

Wang, T.; Wang, G.; Li, X.; Zheng, H.; and Zhao, B. Y. 2013. Characterizing and detecting malicious crowdsourcing. In *SIGCOMM*, 537–538.

Zhang, H.; Lai, J. K.; and Bcher, M. 2012. Hallucination: A mixed-initiative approach for efficient document reconstruction.