ABSTRACT
Harnessing human computation through crowdsourcing offers an alternative approach to solving complex problems, especially those that are relatively easy for humans but difficult for computers. Micro-tasking platforms such as Amazon Mechanical Turk have attracted large, on-demand work force of millions of workers as well as hundreds of thousands of job requesters. Achieving high quality results by putting humans in the loop is one of the main goals of these crowdsourcing systems. We study the effects of different user interface designs on the performance of crowdsourcing systems. Our results indicate that user interface design choices have a significant effect on crowdsourced worker performance.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces– Graphical user interfaces, Interaction styles.

General Terms
Performance, Design, Human Factors

Keywords
Crowdsourcing, User Interface, Cognitive Load

1. INTRODUCTION
Crowdsourcing has been discussed under various labels, including open innovation, collective intelligence, human computation, mass collaboration and distributed problem solving, among others. It involves the harnessing of the collective knowledge and intelligence of a large number of individuals to generate solutions to relatively complex problems. Crowdsourcing has been reported to be efficacious in a variety of problem solving activities and turns out to be more effective than purely computational approaches for some classes of problems. Human inputs are acquired and aggregated over the internet for solving problems or aspects of problems that are relatively easy for people but difficult for computers, especially in areas such as image analysis, speech recognition, and natural language processing.

The Amazon Mechanical Turk (AMT or MTurk) and Crowdflower are examples of platforms that implement microtask-based crowdsourcing. These enable problem requesters to contract and interact with an on-demand, global workforce through a web-based user interface. Monetary reward seems to be the main incentive and workers try to earn as much money as they can in short periods of time. Job requesters have to accept and pay workers for results or reject workers’ results without paying them. In the case of tasks which solicit people’s opinions, it is not possible to check all responses from workers and reject low performance results. Requesters tend to be willing to pay more for high quality crowd inputs. From both research and practical perspectives, it is important to identify the critical factors that affect crowdsourcing system performance and to create interfaces that can promote improved worker performance.

A good proportion of the previous research in this area has tended to focus on factors that affect the motivation and creativity of workers and on cheating detection methods. There have not been many studies that deal with the impact of visual design of the tasks’ interface on workers’ performance. The usability of the software and user interface that are part of platforms such as MTurk can potentially affect worker satisfaction and performance levels and, by implication, the overall costs incurred by the requesters.

While many researchers have studied usability of systems in software design [9, 12, 16], and the effects of cognitive load and its integration with human computer interaction (HCI) concepts on user interface design [1, 8], there are few studies that have addressed the effects of user interface design in the crowdsourcing context. User interface design acquires even greater significance for crowdsourcing tasks given that the tasks are performed by a large number of globally dispersed. We argue that the design of the interfaces through which the workers perform human computation and related tasks has a significant effect on the performance. In order to test this, we designed an experiment to investigate the effects of different user interface designs on performance of a crowdsourcing application. The details of the design and the results are presented below.

2. BACKGROUND AND RELATED WORK
The phrase ‘crowdsourcing’ was originally coined by Howe who defined it as “… the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call” [3]. It has evolved over the years into a range of endeavours including open innovation, distributed human computation, prediction markets, crowdfunding, and crowdservicing, to name a few [4]. We focus on the latter in which complex tasks are broken down into a large number of microtasks which are assigned to a large number of online workers through platforms such as Mechanical Turk.

In Amazon Mechanical Turk (MTurk), requesters can post their tasks. Workers sign onto the system, search for their preferred tasks, accept and solve the tasks and send the results back to MTurk. Micro-tasks on MTurk are referred to as HITs (Human Intelligence Tasks) and are grouped into HITGroups. Each requester can assign the same HITs to more than one worker. Some HITs are just single tasks while others can be a collection of micro-tasks, such as providing labels for images or annotating text.

MTurk provides web-based interfaces for both requesters and workers to implement microtask-based crowdsourcing. Workers
can login, search for tasks and perform the job. Requesters can choose between the web-based user interfaces (UI) to create simple HITs and collect results, or create more complex HITs using specialized UIs using Amazon’s API for MTurk. MTurk APIs support a variety of programming languages.

Regardless of the approach used to create HITs (web UI or API), all tasks are shown in an iframe inside workers’ main web interface page. In order to view the HIT and complete the task, workers need to scroll within this iframe. This limited HIT design environment highlights the importance of a good UI design which has the potential to affect the quality of results provided by workers, as posited by HCI theorists [7].

2.1 Crowdsourcing Worker Performance
Since workers’ performance influences overall crowdsourcing system performance, several researchers have studied aspects of worker performance. Monetary incentive is the primary motivation in MTurk. A few studies have investigated the effect of increasing the reward to get higher quality results. Franklin et al. studies [7] showed that increasing monetary rewards decreased the time required for the HIT to be picked by workers. Another study by Faridani et al. [6] implied that the demand for tasks increased as a result of increased reward. However, in some crowdsourcing applications, higher rewards did not necessarily translate into higher quality responses [2, 7, 13]. Cases have been reported in which increased payment resulted in reduced demand for the tasks, since high reward usually signifies higher complexity tasks [7]. Based on these studies, increasing reward by itself may not necessarily lead to improved results. It is important to assign a proper reward for the HITs in order to achieve lower response time and higher quality results. As well, research by Franklin et. al and Kittur et. al [7, 11] have shown that adding precise instructions to HITs and providing cheat detection mechanisms within the HIT directly have positive effects on worker performance. These mechanisms can also help to discourage gamers who try to cheat.

While the amount of monetary rewards and HIT instruction design can potentially affect workers’ performance, there is not enough research on the impact of HIT UI design on workers’ performance. As mentioned in the foregoing, the main mechanism of communication between workers and MTurk is the HIT’s web-based UI. Khanna [10] studied the impact of usability and UI design of MTurk on low-income workers in India. They discovered that varying skills are needed to do most of MTurk tasks and highlighted the need for designing better mechanisms to match tasks with workers capabilities. They also pointed out that complexities of user interface design and task instructions prevented their target group of workers from completing the task.

3. RESEARCH QUESTIONS AND HYPOTHESIS
As noted before, maximizing performance is a goal in most crowdsourcing tasks. Requesters want their crowdsourcing tasks to be completed in the shortest possible time with high quality results. Our aim is to investigate HITs’ UI design and its impact on crowdsourcing tasks’ performance. We specifically address the following research questions:

- Does the design of user interfaces impact workforce performance and productivity?
- Does the design of user interfaces impact workforce performance and productivity?

One of the CLT design suggestions is eliminating unnecessary and distracting features in the UI. If there are too many unnecessary features in a UI, more of the working memory will be wasted dealing with these features. It has been documented that when unnecessary features are eliminated, user’s cognitive load will be minimized and contribute to higher learning ratio in educational software [15]. In the context of this research we explore this design principle and the effect of diverse UI designs on the performance of workers in a given crowdsourcing task.

The specific hypothesis is:

Lowering extraneous cognitive load by eliminating unnecessary features from HIT UI design will result in higher quality responses from workers.

4. RESEARCH METHODOLOGY AND DESIGN
We describe the design of the experiment that we performed to test the hypothesis. The crowdsourcing task we designed is an image ranking task in which we asked crowd workers to rank ten images based on their similarity to a given query image. The particular task was chosen because it involves non-trivial visual information processing for which the quality of the user interface is particularly critical. For this experiment, we designed three different UIs based on ranking, direct sorting (drag and drop), and rating. The aggregated ranking provided by crowdworkers using each UI was compared to the gold standard for performance analysis. All three UI designs are described in the following sections.

4.1 Rank UI design
For the Rank interface, we provided the workers 10 randomly selected images and asked them to assign a number between 1 and 10 to each image according to their similarity to the given query image. Value 10 means the given image is the most similar image to the query image and value 1 means the image is the least similar image to the query image. Workers have to select a number for each image and they cannot use each value more than once. In this task design users had to compare 10 images with the query image and rank each image not only based on its similarity to the query image but also based on the degree of similarity to other images of the query image.

4.2 Sort UI design
For the sort UI for ranking images, we used some of the JQuery UI functions to create a drag and droppable list of images and asked workers to sort images based on their similarity to the given query image using the drag-n-drop functionality of the HTML page. (Screenshots can be made available)

4.3 Rate UI design
For the Rate user interface design, we provided the 10 randomly selected images and asked them to assign a number between 1 and 5 to each image according to the degree of similarity of each image to the given query image, with 5 for high similarity and 1 for low similarity. The workers had to provide a rating for each image and they were allowed to use the same rating value more than once (ties were allowed).

Unlike the Rank method for which workers had to compare all images to provide the ranking, they were able to focus on each
The ranked list was created based on the average ratings. We calculated the correlation between each of the three aggregated rankings and the gold standard ranking using Spearman’s ρ correlation metric.

### 4.5 Dataset

Our experiment involves assessing the performance of the workers in the ranking tasks for which a gold-standard is needed. We decided to use Corel-Princeton Image Similarity Benchmark Dataset for this reason. This dataset contains eight categories of 30-50 images with a corresponding query image for each. For each query image, the ground truth consists of a rank-ordered list of images based on the degree of similarity with the query image. Aggregated rankings provided by workers were compared against this gold-standard ranking.

### 5. EXPERIMENT: IMAGE RANKING

As part of our experiment we created and posted several HITs using the three UIs (Rank, Sort and Rate) that we designed. The HIT structure for this experiment was:

- $0.05 reward for all three types of HITs
- instructions for users to do the task
- each worker was permitted to perform the task only one time.
- added time stamps to the design of the HIT to identify and remove the workers who just clicked and did not perform the task carefully.
- for each UI design we arranged for 50 different workers to perform the task.

We designed a system to create the HITs, collect and store the results obtained from the workers. MTurk makes it possible for workers to view the HIT in preview mode before accepting it. However in our experiment we just showed a simple preview description and not the full HIT. Each time a worker accepted a HIT, the corresponding page was created on the remote host and the HIT with proper UI showed to the worker to perform the task. Screenshots for each UI design are not included due to space constraints.

Users send their results back to MTurk using “Submit” button we provided on each page and the data collection was performed programmatically.

For our experiment we selected six categories (Airplane, Car, Fruit, Flower, Horse and Model) out of the eight Corel-Princeton categories and randomly selected 10 images from each category. For each selected category we used the HIT creation system and posted three different HITs with the previously described UI designs (Rank, Sort and Rate). For each HIT we asked 50 crowdworkers to perform the task. The total number of HITs created for this experiment was 900 (6category*3UIdesign*50worker) at an overall cost of $45.

### 5.1 Analysis and Results

We aggregated the results for Rank and Sort UI design using Scaled Footrule Aggregation [5] and for Rate UI design we computed the average and sorted the list to create ranked lists. To compare the new rankings with the ground truth we calculated Spearman’s ρ rank correlation between the gold-standard and crowd-generated ranking. Higher the rank correlation coefficient, closer the aggregate ranking is to the gold standard and hence indicative of higher performance.

Our results show that rank correlation coefficient of the results produced using Rate UI design is higher than the other two UI designs. This difference is statistically significant. This implies that the ranked list produced by users from Rate user interface is more similar to gold-standard ranked list and we can conclude that the Rank UI design leads to better performance (Table1).

### Table 1. Distance between gold-standard ranking and experiment1 results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Rank UI</th>
<th>Sort UI</th>
<th>Rate UI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>Spearman’s ρ rank correlation</td>
<td>0.54</td>
<td>0.79</td>
</tr>
<tr>
<td>Cars</td>
<td>Spearman’s ρ rank correlation</td>
<td>0.51</td>
<td>0.84</td>
</tr>
<tr>
<td>Flower</td>
<td>Spearman’s ρ rank correlation</td>
<td>0.66</td>
<td>0.86</td>
</tr>
<tr>
<td>Fruit</td>
<td>Spearman’s ρ rank correlation</td>
<td>0.58</td>
<td>0.59</td>
</tr>
<tr>
<td>Horse</td>
<td>Spearman’s ρ rank correlation</td>
<td>0.80</td>
<td>0.77</td>
</tr>
<tr>
<td>Model</td>
<td>Spearman’s ρ rank correlation</td>
<td>0.12</td>
<td>0.32</td>
</tr>
</tbody>
</table>
image makes the entire list to be rearranged. These movements of the images can distract the user from the original task and impose additional cognitive load on the workers performing the task. As well, the number of user clicks required is higher in this UI design. Unnecessary distracting features and high number of clicks put more cognitive load on the task and tend to poor results from users.

We suggest that the reason that workers perform better with the Rate UI design is they can focus on each image in isolation and assign a similarity score without reference to the other images. This reduces the burden of making comparisons resulting in lower cognitive load. The critical implication of our study is that careful attention in UI design to reducing the cognitive load of crowdsourcing workers can lead to significant performance benefits. Our preliminary results point to the need for more research on UI design for crowdsourcing platforms from the perspective of cognitive load and usability. Cognitive load aspect of HIT design in textual tasks can also be studied. The findings of this and other related studies have the potential to benefit the design of crowdsourcing platforms and to contribute to improved crowdsourcing worker and system performance.

7. CONCLUSION

In light of the limitations of the generic user interface in MTurk, it is important to design HIT UIs that can help reduce worker cognitive load and increase worker productivity. This has the potential to reduce the execution time of the crowdsourcing task. In this paper we tried to study the impacts of user interface design of HITs in the MTurk crowdsourcing platform on workers’ performance. Our experiments show that designing a HIT UI with the goal of reducing the cognitive load will help workers focus on the task and perform better.

We have presented preliminary evidence to suggest that by spending more time on HIT UI design, requesters can achieve improved results. These results can help develop guidelines for making crowdsourcing tasks easier to perform.

8. REFERENCES


