

Wallah: Design and Evaluation of a Task-centric Mobile-based Crowdsourcing Platform

Abhishek Kumar
Xerox Research Centre India

Kuldeep Yadav
Xerox Research Centre India

Suhas Dev
Xerox Research Centre India

Shailesh Vaya
Xerox Research Centre India

G. Michael Youngblood
Palo Alto Research Center

kuldeep.r@xerox.com

ABSTRACT

Crowdsourcing through web technologies has emerged as a key method and tool for conducting distributed work. There are new platforms constantly emerging that aim to provide crowdsourcing opportunities on mobile phones. However, most of these systems are very specific to certain types of tasks and do not address various mobile resource constraints experienced in developing countries such as India.

We propose and design a new platform *Wallah* that tries to address these limitations with a broad vision to make crowdsourcing opportunities pervasively available and feasible to do (i.e., at all times, locations, and with minimum investment in infrastructure from the crowd-workers). It supports task-centric applications to minimize the impact of screen size and facilitates caching of crowdsourcing tasks to deal with network limitations. *Wallah* supports both physical as well as virtual crowdsourcing tasks. The current version of *Wallah* implements an end-to-end platform for Android devices and includes five different task-centric applications for different categories of crowdsourcing tasks (human OCR, image tagging, language translation, audio transcription, and video tagging) developed by us. We evaluated the system with a 2 week pilot deployment among 59 crowd-workers where over 16000 tasks were performed. We analyzed the platform usage in detail and present descriptive statistics related to task completion time and task accuracy rates with other analysis such as the impact of screen size on task completion time and accuracy. We also conducted a post study survey to get participant's qualitative feedback and their perceived difficulties of different crowdsourcing tasks.

1. INTRODUCTION

Crowdsourcing is a methodology for solving certain type of problems (tasks) that are hard to accurately compute even by modern day computer technology, but easy to accomplish by a distributed group of human beings. Currently, there are several mobile and web-based platforms for hosting crowdsourcing tasks such as

Amazon Mechanical Turk (AMT)¹, GigWalk², and Jana³. Most of AMT's workforce is from the US and India [15]. Even though 36% of AMT users are from India, most of them are high end users who have access to desktop computers with Internet.

Mobile-based platforms increase coverage of crowdsourcing to the people who do not have access to traditional web interfaces. Also, they provide functionality to perform crowdsourcing tasks anywhere, any time. The majority of new cellular subscriptions today are coming from developing countries. Also, the number of smartphones in developing countries are increasing at a phenomenal rate. For example, smartphone users are increasing at a yearly rate of 89% in India and specifically, urban smartphone users have doubled in numbers in 2013. In countries like India, China, and Egypt, over 50% of users access the Internet from mobiles only [3]. In essence, enabling crowdsourcing on mobile phones can create opportunities and open new possibilities to a potentially large number of workers, and it may help realize the futuristic vision of agile workplaces. Also, people in developing countries can largely benefit from such a platform as it creates scalable employment opportunities that do not have to be located in urban centres.

Currently, there are only a few initiatives in the direction of enabling crowdsourcing tasks on mobile phones, specifically in the context of developing countries. Some of them are MobileWorks [15], TxtEagle [14] and SamaSource [18]. Among these platforms, TxtEagle provides generic platform capabilities, but it is restricted to SMS as a communication medium, which restricts its usage for certain kind of tasks only. MobileWorks faces similar restrictions as well because it is tailored for feature phones and does not exploit native smartphone capabilities such as sensors. SamaSource does not enable mobile-crowdsourcing, but rather a crowd-worker management platform by establishing outsourcing centres in developing countries.

There is a need for a generic mobile crowdsourcing platform that can engage worker populations in emerging countries such as India. However, building such a platform is challenging due to following reasons.

- The majority of users in developing countries are still using 2G based internet connection. As per recent statistics, the number of 3G subscribers in China are only 14% of the total 1 billion cellular subscribers [5], while in India, it is only 2% of over 893.8 million subscribers [6].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MOBIQUITOUS 2014, December 02-05, London, Great Britain

Copyright © 2014 ICST 978-1-63190-039-6

DOI 10.4108/icst.mobiquitous.2014.258030

¹<https://www.mturk.com/mturk/welcome>

²<http://gigwalk.com/>

³<http://www.jana.com/>

- WiFi does not exist in the majority of areas, and when it does it is often not public. Cellular Internet provides limited download speed with frequent download failures [13].
- Some of the crowdsourcing tasks are challenging to perform in the limited screen sizes that smartphones offer. There is a need for interfaces that can overcome this limitation.
- There exists large amount of device heterogeneity where the devices vary in terms of cost, capabilities, screen sizes, etc. For example, the cost of a smartphone in India ranges from \$70 USD to \$500 USD. This kind of heterogeneity makes it challenging to design a generic crowdsourcing system.

We take the above challenges in consideration and designed a mobile-based crowdsourcing platform *Wallah* that provides a generic as well as task-centric interface for crowdsourcing tasks. *Wallah* has three different components: 1) a Cloud instance, 2) the core mobile application, and 3) task-centric applications. The Cloud instance controls all the operations of the Wallah platform. The mobile application is used by the task requestor to submit tasks and the crowd-workers to perform tasks. Every task in Wallah platform is associated with a task-specific application, which is designed to increase the crowd-worker efficiency (i.e., minimizing the time to complete a task) and quality of task performed. The task requestor can either develop a task-specific application or they may use one of the existing application templates when posting a task. Additionally, Wallah provides open APIs to integrate its functionalities with other third party apps that may need crowdsourcing for some of its operations.

Specifically, this paper makes following contributions.

1. We designed a generic mobile-based crowdsourcing system that can support a varied number of crowdsourcing tasks. The task-centric architecture of Wallah supports different categories of tasks (i.e., virtual tasks such as Human OCR, physical tasks such as point of interest data collection) as well as it supports real-time crowdsourced question-answering services. The Wallah platform is specifically designed to address the challenges that are faced by mobile users in developing countries while performing crowdsourcing tasks. It supports task-centric applications to minimize the impact of screen size and facilitates caching of crowdsourcing tasks to deal with network limitations.
2. We implemented an end-to-end platform for Android devices. Also, we designed and developed five different task-centric applications for different categories of crowdsourcing tasks using the Wallah platform. The categories of tasks include human OCR, image tagging, language translation, audio transcription, and video tagging. These applications use specialized interfaces to handle different screen sizes and are designed to minimize task completion time.
3. We deployed the Wallah platform with five task-centric applications in the real-world by recruiting 59 crowd-workers for a 2 week study. In total, crowd-workers performed 8512 tasks as well as 8342 validations using the proposed platform. We analyzed the platform usage in detail and present descriptive statistics related to task completion time, task accuracy rates, and the worker’s work patterns. Furthermore, we evaluated the impact of screen size on task completion time and the quality for different categories of tasks.
4. We conducted a survey of the crowd-workers who participated in the Wallah deployment study to gather the user’s

qualitative feedback, perceived difficulties of different crowdsourcing tasks, and further scope of improvements in the platform.

While the *Wallah* platform is designed to improve the efficiency of both crowd-workers and task requestors, in this paper we performed a focused evaluation of the *Wallah* platform only from crowd-workers’ perspective. The evaluation with respect to the task requestor will be performed in a subsequent user study.

2. RELATED WORK

In this section, we outline the existing mobile-crowdsourcing systems and platforms. There is no dearth of crowdsourcing platforms in the technology space, but only a few of them have been able to exploit mobile capabilities for crowdsourcing. The concept of crowdsourcing gained significance in the mobile domain with the evolution of TxtEagle [14], which is a SMS based mobile crowdsourcing marketplace deployed in Kenya and Rwanda for tasks like translation, transcription, and surveys etc. The system is dependent on SMS as a medium of communication and hence, its capabilities are limited to only a certain set of tasks.

Human OCR tasks were induced into mobile-based crowdsourcing with MobileWorks [15]. However, it is a mobile web based system and does not use the native features and capabilities of a smartphone. MClerk [16] is another system that uses images via SMS (or MMS as it is referred to in the US) approach to distribute and conduct digitizing tasks for English as well as Non-English languages. More sophisticated tasks like image collection, image tagging, and road traffic monitoring were made feasible on mobiles by mCrowd [17], which is an iPhone based mobile crowdsourcing client for ubiquitous sensing tasks. It utilizes the native features of the phone and at the same time exploits the existing crowdsourcing platforms like AMT, ChaCha, etc. for posting tasks. However, it is not a stand alone mobile crowdsourcing platform and acts like a proxy between mobile and existing crowdsourcing platforms.

In a user study conducted by Della *et al.* in 2013 [19], they found the current crowdsourcing platforms inadequate for mobile devices. Their study was centred on comparing the performance of various tasks (content categorization, moderation of an image, image tagging, transcription, etc.) on mobile devices with the performance of tasks on desktop. In [11], Finnerty *et al.* studied the impact of variable rewards and the complexity of interfaces on the quality of crowdsourcing tasks and suggested that simple interfaces lead to good performance. Musthag *et al.* [20] analyzed the labor dynamics in the mobile micro task market space for a physical crowdsourcing based platform with respect to factors such as demographics and data quality.

3. SYSTEM DESIGN AND IMPLEMENTATION

Wallah is designed as a generic platform that makes it easier for a task requestor to create crowdsourcing tasks. The task-centric model of Wallah enables crowd-workers to complete tasks efficiently with the flexibility of doing them anywhere, anytime. Additionally, Wallah provides interfaces to integrate third party applications while supporting both virtual and physical world tasks. Wallah has three different functional components (i.e., a cloud instance, a mobile application, and task-centric applications) as shown in Figure 1. In this section, we will describe these components as well as the interactions among them in detail.

3.1 Cloud Instance

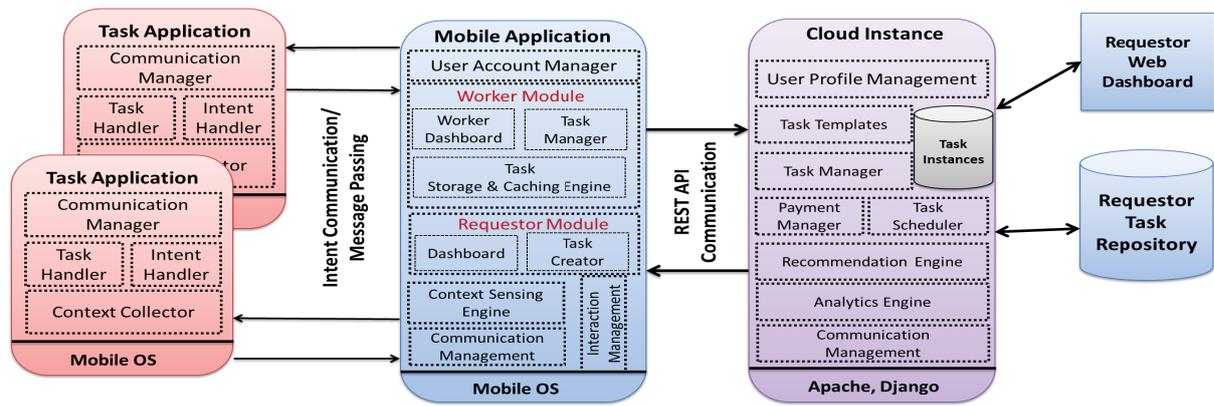


Figure 1: System design of *Wallah* platform with different functional components.

The Cloud instance in the *Wallah* platform hosts all of the key modules and serves as a controller for the mobile application. In this subsection, we provide a brief description of these modules.

- **User Profile Management:** It handles the creation, modification, and removal of user profiles, which includes both crowd-workers and task requestors. This module is responsible for the authentication of all the users on the platform. This module also stores the previous task history for every user, demographic details (i.e. age, gender), location, competencies, reputation of the user, and user preferences.
- **Task Templates:** This module stores the templates for common crowdsourcing tasks such as image tagging, Q&A, audio transcription, etc. These templates can be accessed by the task requestor using the mobile application or a web-based dashboard to create new tasks quickly. The task requestor also has flexibility to modify these templates to accommodate their needs.
- **Task Manager:** This module is responsible for creating and maintaining task instances when a set of tasks are submitted by the requestor. It manages the whole life cycle of task instances from their submission, assignment, intermediate status, collection of responses, and validation as well as their final delivery back to the requestor. The task manager also maintains task profiles with the mapping and allocation of tasks to the workers (task-worker profiles). All the task specific data between task repository and the mobile application is tunneled via this module. The task responses data is stored back to the task repository through this module while updating the task, task-worker, and user profiles.
This module is also responsible for creating the task-specific workflows for task validations based on the requestor requirements. For instance, a requestor may want a task response to be validated from two different independent crowd-workers to check the correctness. The task manager automatically embeds these requirements while creating workflow for a certain category of tasks.
- **Payment Manager:** This module interacts with the user profile and the task manager to compute the crowd-worker payment based on task completion. The task requestor can access this module through a web-based dashboard or mobile application that provides them functionality to release the payment to crowd-workers. Subsequently, it credits the crowd-workers' account.

- **Recommendation Engine:** It generates task recommendations to the crowd-workers based on their user profiles and available tasks on the platform. The task recommendations are based on several parameters such as context, location, worker competencies, previous task history, and performance. This module assists crowd-workers in selecting the appropriate tasks, which best matches their profile and capabilities. For example, Alice as a task requestor posts a translation task to convert Hindi text into English. This task will be recommended to a crowd-worker, who is proficient both in Hindi and English.
- **Task Scheduler:** All of the enterprise crowd-sourcing tasks are bounded by a SLA (service level agreement) deadline. However, crowdsourcing platforms have a dynamic nature and there is wide variance in terms of task completion time, quality, and completion rates [9]. This module provides tracking capabilities as well as recommendations to requestor in intelligently scheduling tasks, so that the overall turn around time is optimized as well as the quality is improved. Typically, these recommendations are derived from historical data as well as by scheduling a small number of tasks to learn the current behaviour of the platform.
- **Analytics Engine:** The platform logs each and every interaction of the requestor as well as the crowd-worker on both the mobile application and the cloud instance. These logs are analyzed to create a detailed analysis and recommendations for the task requestor. For example, if Alice as a task requestor posts a few jobs and the number of crowd-workers who are completing the tasks are very few, fine-grained analytics on the platform logs can help Alice in understanding and improving task designs.
- **Communications Management:** All the communication between the mobile application and the cloud instance is accomplished by this module. The Cloud instance exposes REST based APIs for enabling this communication channel. This module makes it simple to interface with other third party APIs. For instance, a task requestor has the flexibility to upload all the task instances on the *Wallah* platform or to provide a URL/APIs from which tasks can be fetched. This module has the capabilities to interface with the web-based requestor dashboard.
- **Requestor Web Dashboard:** It is web based interface between the task requestor and the different modules of the

Wallah platform. It enables the requestor to create new tasks and also provides the capabilities to modify/remove any existing tasks. The requestor can use existing task templates to create task instances. They can also modify the task templates as per any defined requirements. The requestor can upload the task-specific data from the Wallah task repository or can specify communication APIs to their own task repository. Using the dashboard, the requestor can monitor the status as well as the responses of tasks. It also provides other functionalities such as profile management, analytics, visualization, etc.

3.2 Mobile Application

The Wallah mobile application serves as an interface to different modules provided by the cloud instance. It allows the task requestor to create tasks and receive responses from crowd-workers in real-time. The mobile application is a focal point for the crowd-workers and it can be used to find available tasks, select a few of them, launch task-centric applications, and manage their profiles. Next, we provide a brief description of the different modules of the mobile application.

- **User Account Manager:** This module is responsible for user login and signup on the platform. It allows a new user to register as a worker or as a requestor and provides interfaces to edit or modify their profile or preferences. Additionally, this module also provides the user a summary of finished tasks and payment details.
- **Workers Module:** Whenever a user logs into the mobile application, the user account manager classifies them as a crowd-worker or requestor. If a user is identified as a crowd-worker, this module is initiated. Mainly, it provides following functionalities.
 1. **Task Manager:** The Task Manager module fetches the available tasks from the cloud and presents a list to the worker. They can see the task details such as the description, payment information, expected time to complete, expiry time, and so forth for every task available on the platform. Whenever, a user accepts any task, the task-worker instance is created at the Cloud, as well as task data and task specific application data is made available to the worker. The Task Manager is also responsible for the installation and uninstallation of task specific applications from a users' device with their consent whenever the associated tasks are accepted or expired respectively.
 2. **Task Storage and Caching Engine:** This module is responsible for two main scenarios, first to store and cache the task related data locally until the task is solved by the worker and second to store the solved task data until it is synced back to the Cloud. After accepting a task, the crowd-worker has the flexibility to cache a task till some time t . The time t is a configuration parameter on the platform and the requestor can set it while submitting a task. With the help of the caching functionality, the user can accept, fetch and store tasks whenever Internet connectivity is available to them, and then can solve the tasks later on (even without Internet connectivity). The task responses data is also locally stored on the mobile device until an Internet connection is available. Upon availability of an Internet connection, this module then uploads the solved task data

onto the server and then marks the completion of the task. In case of delayed submission, the Wallah platform and requestor has the right to reject the responses submitted by the crowd-worker.

3. **Workers Dashboard:** It displays the various tasks that has been accepted by the crowd-worker as well as the other available tasks. The dashboard also displays the battery, storage, and network information of the mobile device. This dashboard is also used to launch a task-centric application whenever a crowd-worker wants to perform a certain pending task.
- **Requestors Module:** Mobile application can be used by requestor to post tasks as well as to track status of already submitted tasks. After login, if a user is identified as a requestor, this module is initiated to facilitate the platform usage for a requestor.
 1. **Task Creator:** The Task Creator gives the functionality to the requestor to post new tasks onto the server, the requestor is required to give the task details including description, payment information, time to complete, expiry, the link to requestor repository API (if opted for), and the selection of a template to connect a task-specific application. The task created is then available for the workers to accept and solve.
 2. **Requestors Dashboard:** It displays the tasks that have been created by the requestor. The requestor has the flexibility to track, change, modify, or remove an already existing task. In case of small tasks, the requestor can use the platform to visualize task responses and release payment for the crowd-workers accordingly. More complex task assessment and payment release can be performed on the web.
 - **Context Sensing Engine:** Contextual information is one of the central aspects of many physical crowdsourcing applications. This module monitors the users' data streams and sensor data (e.g., user location, GPS, WiFi) as well as communicates directly with the users to determine their current context. In general, *Wallah* uses triggered-sensing to save energy, whenever possible. For example, it uses low energy interfaces i.e. GSM to sense place-based information and occasionally, switches on WiFi to improve accuracy [22]. The inferred or known context is shared within the system for all other components to utilize in order to provide contextually relevant information, which could be used for task offering/solicitation.
 - **Communication Management:** This module ensures the communication of the mobile application with the cloud instance and the inter-application communication with the task-specific applications. REST APIs are used for the communication with the Cloud and message passing interfaces using intents in Android OS are used for communicating with task-specific applications.

3.3 Task Specific Applications

Wallah provides the unique functionality of associating every task with a stand alone mobile application. One rationale behind allowing task-specific applications is that they provide greater freedom for application developers in designing specialized interfaces for task efficiency and quality. Furthermore, task-specific applications can make use of native mobile OS features such as sensors

etc. In this subsection, we briefly discuss the different modules of task-specific applications.

- **Communication Manager:** This module builds the communication channel between the Wallah mobile application and the task-centric application using the message passing interfaces of the mobile OS. This module can also be used to access third party web services such as a speech-to-text service.
- **Intent Handler:** It works in conjunction with the Communication Manager module to receive and handle intents or broadcast messages from the mobile application.
- **Task Handler:** This module extracts the task data from messages and displays it on user interface accordingly. It also controls the whole workflow of a task request.
- **Context Collector :** This module interfaces with the mobile OS to collect sensor data such as accelerometer, light, etc.

3.4 Implementation Details

Wallah has been implemented as a JSON-based cloud service where we expose the APIs including the authentication APIs along with the rest of the communication APIs. The Wallah Cloud has been implemented using Gunicorn, Python, and Django and runs over Ubuntu OS.

Our aim is to exploit the capability and usability of a smartphone to its fullest, and therefore, we chose to develop native mobile applications over mobile web-based applications. Currently, due to wide-spread adoption of Android devices in developing countries such as India, we developed the initial version for Android devices only. However, the APIs from the web service are generic and can be used for other native platforms as well as cross-platform applications. Currently, we support nearly all Android devices with version 2.2 and above ranging from smartphones to tablets. The system supports task-specific applications, which are also built with compatibility for Android 2.2+ devices. These applications interface with Wallah mobile application using intents.

Figure 2a shows the login screen of the mobile application, which has options for worker and requestors to log into the application. Figure 2b shows the application dashboard, which contains the list of jobs accepted by the crowd worker, battery usage, and memory usage. The users can observe how much battery and memory consumption is made after performing the task. Figure 2c shows the payment details screen which has the details of the incentives and rewards earned by the crowd worker.

3.5 Use-case

The following is an instance of how the Wallah platform is used by a task requestor and crowd-workers.

1. The task requestor logs in to the task requestor dashboard to create a new task.
2. For creating a new task instance the requestor has two options. First, to use an existing task template and modify its parameters as per their requirements in order to create new tasks quickly. Second, to create a new task template (built on the top of standard guidelines published for the requestor), upload it to the cloud instance, and then create a task instance.
3. The requestor needs to upload the task specific input data for the task instances on Wallah to the task repository or they



(a) Wallah Login (b) Wallah Dashboard (c) Wallah Payment

Figure 2: Screenshots of Wallah Android mobile application

can specify communication APIs of their own task repository. Once the input data associated with the task instance is available, the set of tasks are available for crowd-workers' acceptance and solving.

4. The crowd-worker logs in to the Wallah Mobile application to see a number of tasks available. They can see the details of the tasks including the description, time, incentives, etc. as shown in Figure 2b.
5. The crowd-worker accepts the task and the task will be available in their dashboard. They can then fetch the task related data and solve it. They also have the flexibility to cache the fetched data for solving tasks later.
6. The worker launches the task-specific application from the dashboard to solve the task
7. The worker submits their response for the task, as soon as the response is uploaded to the Wallah cloud the task completion is marked and the results are available to the task requestor.

3.6 Task-Centric Applications for Different Category of Tasks

We have built the following proof-of-concept task-centric applications that use the Wallah platform. Each of the task-centric applications receives task data from the Wallah mobile application and submit the result back after completion.

1. **Human OCR Task Application:** There are many hand-written documents that need to be digitized for processing. Typically, automatic OCR engines do not work on many of these documents due to reasons such as poor handwriting. As part of the Wallah platform, we designed a mobile application that can help in digitizing such forms with the help of crowd-workers. As shown in Figure 3a, it provides an image to the crowd-worker and they have to determine and write the content of the image in the given text box. The crowd-worker may also choose to use the specialized interface of the app (i.e., speech-to-text input) to avoid typing. The application presents the top- k phrases after a speech-to-text conversion to the crowd-worker so that they may select the closest one to submit or edit then submit.
2. **Image Tagging Task Application:** An image can contain many objects, which need to be identified separately. The

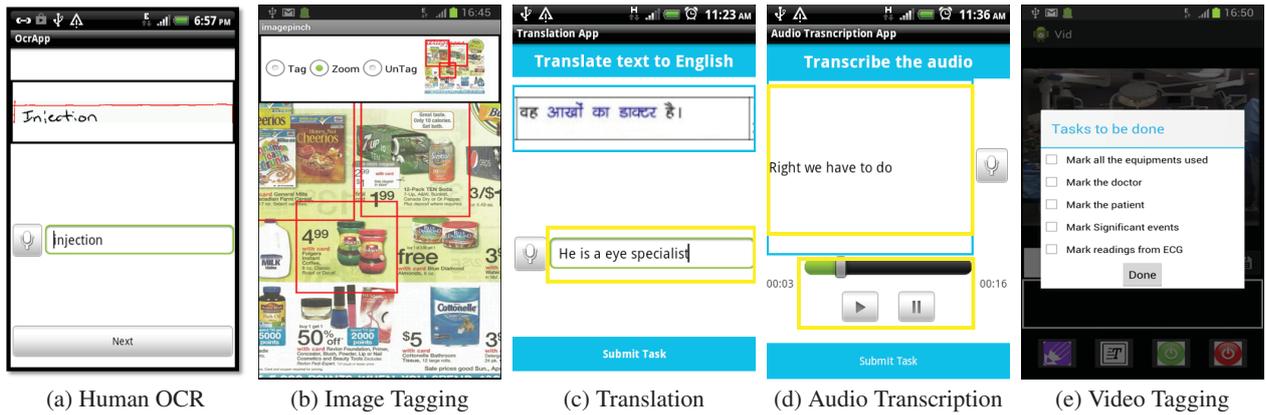


Figure 3: Various task-centric apps of Wallah platform

task is to identify boundaries of these objects, which can be used for different purposes (i.e., to validate output of automatic object segmentation algorithms, or as an input to object recognition algorithms). Mobile screen size is limited and hence there is a need of designing specialized interface, which can help in tagging objects in an image. We designed an application that allows crowd-workers to tag different objects in an image as shown in Figure 3b. The application provides functionality such as pinch and zoom to traverse the whole image. In a large image, a user may lose sight between tagged and remaining to be tagged items. The application provides a birds' eye view (top-right in Figure 3b) that can be used to track the progress of tagging.

3. **Translation Task Application:** There are many applications such as Google Translate, which can be used for translating written text from one language to another. However, it does not work for many languages that are prevalent in Asian countries and often tourists find this inconvenient. Using the Wallah platform, a task requester can post an image of the text that needs to be translated. This image is automatically routed to a crowd-worker who has knowledge of both the source and target languages. Figure 3c shows a snapshot of the application, which is used by the crowd-worker to perform translation. The image shown contains text from a source language and the crowd-worker needs to write the translated text for the target language. Speech-to-text can also be used in this interface.
4. **Audio Transcription Task Application:** Many applications/systems need transcription of audio content. For example, enterprises need to transcribe call center conversations. There are many systems that perform transcription automatically such as YouTubes' automatic captions feature [1]. However, most of these system are not very accurate due to various reasons such as background noise, lack of language models, etc. Using the Wallah platform, a task requester can post an audio file that needs to be transcribed. It automatically splits the audio content into short duration snippets, which are then routed to crowd-workers accordingly. Wallah uses off-the-shelf tools to perform audio splitting [8]. Most of these tools use silence detection approach to split a big audio file into smaller chunks. Crowd-worker use the application interface shown in Figure 3d to transcribe the given audio snippet.

5. **Video Tagging Task Application:** Video event detection is required by many applications/services where the goal is to identify the specified spatio-temporal patterns in a given video. Many of these events cannot be identified by automatic algorithms. For example, an educational video application may need to bookmark a certain part of video where the instructor uses a blackboard. In another scenario, a doctor may need to know a part of an ICU video where a nurse takes ECG readings for a patient. A task requester can use the Wallah platform to crowdsource the tagging of significant events or activity in a given video. In the context of this paper, an event is associated with a single timestamp where an activity could be a time session in a given video. The video is divided into short snippets, which are routed to crowd-workers with a description of the events to be identified as shown in Figure 3e. The application provides an interface to the crowd-worker where they can play, pause, or rewind a video as well as tag mentioned events. The application also provides the flexibility to tag an object in any given frame of the video. Apart from events, a crowd-worker can also mark an activity by marking its start and end time. All the marked events and activities are bookmarked on a parallel video timeline for quick navigation and validation.

Apart from the above task-centric applications, we have designed separate validation applications for each category of tasks. The validation applications also interface with the Wallah platform in a similar way as the task-centric applications.

4. PLATFORM EVALUATION

The effectiveness of any crowdsourcing system can only be assessed in real-world settings. Typically, crowd-sourcing systems are evaluated based on their *task accuracy rate* (i.e., the number of correctly performed tasks vs. total tasks) and *task completion time* (i.e., the time taken to perform different tasks) [15]. To this effect, we framed the following research questions that need to be answered from the evaluation.

- *R1*: What are time requirements to perform different tasks on the Wallah platform? Does completion time vary with respect to the complexity of different categories of tasks?
- *R2*: What are the task accuracy rates of different tasks performed using the Wallah platform? How much does the task quality vary across different categories of tasks?

- *R3* : How does automatic validation by crowd-workers compare to manually validated tasks by the requester?

As all the tasks are performed on mobile phones and many of our task-centric apps contains specialized interfaces to increase task accuracy rates and reduce task completion time, we would additionally like to answer following questions.

- *R4* : What is the impact of different screen sizes on the task completion time across different categories of tasks?
- *R5* : What is the impact of different screen sizes on the task accuracy rate across different categories of tasks?

Moreover, previous research has shown varying task completion rates and accuracy across different days and times on existing platforms such as Amazon Mechanical Turk. We are also interested in the broader observations such as the preferred day/time of crowd-workers to perform tasks, bandwidth consumption among different categories of tasks, etc. To answer these questions, we consider two different evaluation strategies. First, we deployed the proposed platform and associated task-centric apps with 59 crowd-workers spread across 2 cities in India (i.e., Delhi and Bangalore). We used this data to understand platform usage and perform quantitative analysis to answer some of the questions above. Second, we conducted a post study survey to gather users’ qualitative feedback, and the future scope of improvements for the platform. Out of total 59 participants, 33 participants completed the post study survey.

Payments and rewards are integral part of existing crowdsourcing platforms and they significantly impact the response time and quality of task performed [10]. We wanted to keep payments simple and effective. *Wallah* uses a fixed payment scheme where worker was paid for every task performed and payment for each task varied with the complexity. The workers were paid only if their submission met defined quality i.e. 75% in our case. Our post study survey confirms that crowd-workers were fairly satisfied with the payment scheme.

4.1 Deployment

The crowd-workers were recruited using convenience sampling and the only recruitment criteria was the availability of their own personal Android phone. We bootstrapped the platform by creating tasks of 5 different categories ourselves, which were comprised of human OCR, image tagging, translation, audio transcription, and video tagging as discussed in Section 3.6. The crowd-workers were asked to download the Wallah Android application, where they could see all the available tasks and associated task-centric apps. The Wallah platform was configured to obtain single validation for every completed task by a different crowd-worker. We manually created the ground truth for all the tasks to verify accuracy of the completed tasks as well as their respective validation.

During the deployment, we logged several parameters on both the Wallah mobile application and the cloud instance, these include the timestamp at which task was completed, the total time taken to perform the task, the phone model, the OS version, and total data transferred in performing the task among others. We have also logged the fine-grained interaction of the crowd-worker while using the task-centric apps such as the usage of specialized interfaces.

4.1.1 Descriptive Statistics

The field deployment was carried out over a period of 2 weeks and in total, there were 8512 tasks as well as 8342 validations performed by all of the participants. Table 1 presents the task count of individual categories as well as the validations tasks finished during the deployment duration.

Task Category	No of Tasks	No of Validations
Human OCR	4975	4852
Image Tagging	718	709
Translation	1684	1674
Audio Transcription	766	758
Video Tagging	369	349

Table 1: Number of tasks and validations performed for different categories

Crowd-workers used different device models, varied screen sizes, and different OS versions for performing tasks in our deployment. In total, we found that there were 48 different phone models and 18 different screen sizes. Interestingly, the minimum screen size was 2.8 inches and the maximum screen size 8 inches. Also, these phone were running nearly all of the possible Android versions from Froyo (2.2) to Kitkat (4.4.2) [4]. As hypothesized earlier, it is true that there exists a large amount of device heterogeneity in developing countries such as India.

4.1.2 Work Patterns

Here, we analyzed the aggregated working pattern of the crowd-workers (i.e., the number of tasks performed by crowd-workers, preferred day of the week for performing tasks, and the preferred time of the day). As discussed earlier, the Wallah platform collects and maintains these statistics for every crowd-workers in order to intelligently schedule tasks and to maximize efficiency.

Figure 4a presents an analysis of task performed by the top users. Similar to social networks, we observe a power-law distribution where some of the top crowd-workers are very active on the platform and perform the majority of the tasks and others participate/contribute intermittently. For instance, the *top* – 10 users on our platform performed 43% of the total tasks.

We observed that most of the tasks (nearly 77%) were performed on the weekend. This is understandable because most of the crowd-workers who participated in our deployment had regular work (i.e., job, study) during the week. Furthermore, we analyzed the platform logs to find the preferred hour of the day to perform tasks. While individual crowd-workers had their own preferences, most of the tasks were performed in the late evening and night time.

4.1.3 Task Completion Time

Task completion time is an important metric to measure the efficiency of a crowd-worker. In our context, we measure the task completion time as the modulo difference between task start and task end time. The task start time is the time at which the task was presented to the crowd-worker and the end time is when they submitted the task. In our data, we observed few cases where crowd-workers took an unusually large amount of time to submit a task. This could be a case of distraction while performing the task. For instance, in the MobileWorks [15] study, one of the participants admitted doing tasks while watching TV. Hence, we consider *median task completion time* for comparison across different task categories to avoid outliers.

As shown in Figure 5, crowd-workers finished the human OCR task the fastest with a median task completion time of about 10 seconds as compared to the audio transcription task that took the longest with nearly 120 seconds. Audio transcription took the longest time because crowd-workers had to frequently pause the audio player to write the corresponding text. Image tagging tasks also took considerable amount of time for completion due to the tagging of many objects in an image and the frequent zoom-in and zoom-out operations. We can conclude that the crowdsourcing tasks that are more

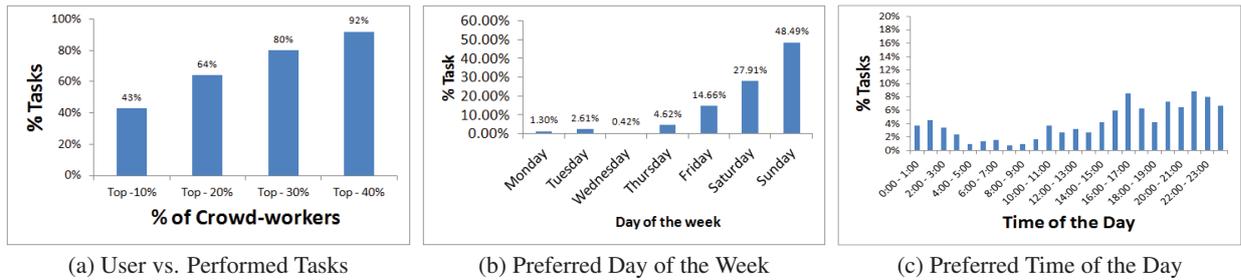


Figure 4: Aggregated working pattern of crowd-workers in Wallah platform

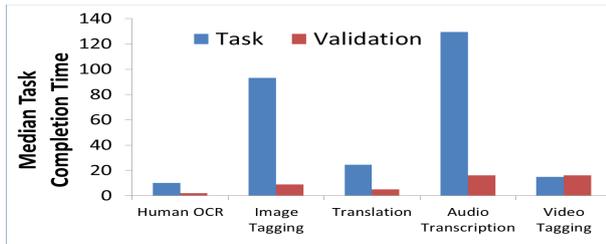


Figure 5: Comparison of median task completion time across different task categories.

interactive than others take more time to complete.

However, we observed that crowd-workers were very quick to complete video tagging tasks. This could happen due to two main reasons; first, video tagging tasks had less typing overhead because these tasks tended towards simple visual information spotting. Second, we found that crowd-workers frequently fast forwarded the video snippets to spot the objects, which may have minimized the overall task time.

In validation tasks, crowd-workers have to agree or disagree on the original task performance. This takes significantly less time compared to performing the actual tasks as shown in Figure 5. Surprisingly, we observed that validation time is similar to the actual task solving time in the case of the video tagging. One of the reasons for this could be that a crowd-worker has to watch the whole video again to spot inaccuracies.

4.1.4 Task Quality

Quality/accuracy is measured as the total number of tasks that are done correctly by the users. Some of our tasks could be subjective in nature. For instance, translation output may be different for different crowd-workers. In such tasks, we ask the crowd-worker to rate the original task performance on a five point Likert scale. To measure the accuracy of task quality, we manually measured the quality of all solved tasks.

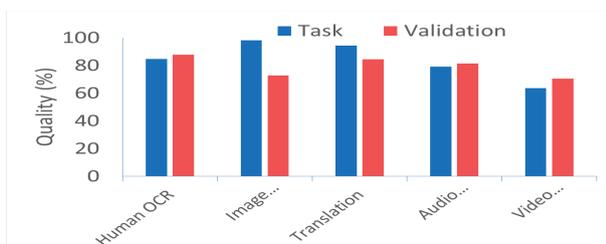


Figure 6: Comparison of task quality across different task categories.

As shown in Figure 6, the highest quality was observed by image tagging task where nearly 98% of tasks were accurately performed by the crowd-workers. We were surprised to note the relatively lower quality (nearly 84%) in human OCR tasks. However, we found that this is due to the poor quality of images supplied where some of the images did not have clear text. Notably, the quality decreased with complex tasks such as video tagging where we found only 63% of the performed tasks to be accurate. We also wanted to understand the quality of the validation tasks. Hence, we measured whether manual tagging is in agreement with crowd-worker based validation. We observed that image tagging tasks had a very poor agreement of only 72% in which crowd-workers were more strict in giving poor ratings to the original task than the manual tagging.

4.1.5 Impact of Screen Size

Here, we analyzed the impact of the screen size on the task quality and the time taken to complete the tasks. Screen size can be considered as an important metric as it can greatly influence the quality of work and the task completion times. For our analysis, the screen sizes were grouped into different categories as follows.

Category Identifier	Screen Size (inches)
C1	< 3.5
C2	3.5 – 4
C3	4 – 4.5
C4	4.5 – 5
C5	> 5

Table 2: Grouping of screen sizes categories

As shown in Figure 7, the crowd-workers who used mobile phones belonging to category C3 took less time in performing all the tasks when compared to the users with phones belonging to other categories. One of the interesting insights that can be drawn from this analysis is that smartphones with bigger sizes do not necessarily reduce the task completion time.

The impact of each screen category on the task quality was observed. As described earlier, the baseline for task quality was created manually. We found that varying screen sizes do impact the task quality as shown in Figure 8. Due to the simplicity of the human OCR task, the screen size had less impact on it. However, audio transcription was negatively impacted the most due to screen size. During transcription the crowd-worker has to frequently pause/play the audio content and subsequently write quite a bit of info into the given textbox, which is affected by the screen size. Surprisingly, C3 was found to be most suitable screen size for the three different tasks (i.e., human OCR, image tagging, and video).

4.1.6 Usage of Specialized Interfaces

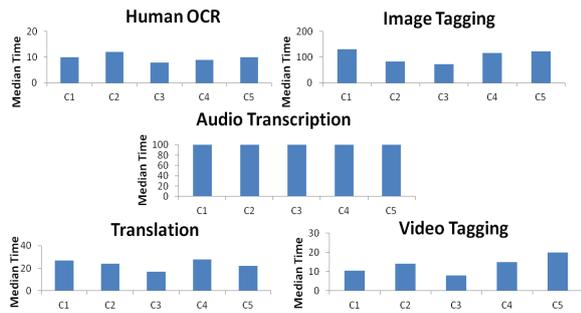


Figure 7: Impact of different screen sizes on median task complete time for different task categories

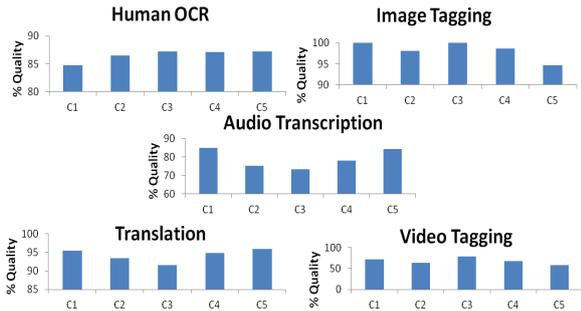


Figure 8: Impact of different screen sizes on task quality for different task categories

The task-centric apps presented in Figure 3 use specialized interfaces to improve user experience of crowd-workers as well as address limitations like small screen sizes. Human OCR, translation, and audio transcription apps had speech input for faster task completion time and it was observed that nearly 60%, 43%, 36% of the crowd-workers used the speech-input based specialized interface one or more times while solving these tasks respectively. However, we found that using speech input increases the task completion time. For example, the median task completion time for Human OCR task was 10 seconds but it increased to 16 seconds with speech input. This is primarily due to two main reasons; the crowd-workers were very efficient in keyboard typing and secondly, speech APIs takes some time in providing results which gets added into the task completion time too. However, in post study survey, it was interesting to note that only few crowd-workers (5%) said that speech input did not save any time for them, where as majority of them were not sure or agreed that it saves time for them.

Image tagging application had a specialized interface with a bird's eye view including zoom and tag functionality. We found that this feature was used by nearly 61% of all image tagging tasks solved on the platform. Crowd-workers pertaining to all the phone categories mentioned in Table 2 made use of the this interface to tag images. We observed that majority of crowd-workers in C1 category relied on this feature more due to small screen sizes of their phone and took its help in nearly 75% of the tasks.

4.2 Post Study Survey

We conducted a post study survey to gather users' qualitative feedback, and to scope future improvements for the platform. We emailed the survey form to all of the participants, and got responses from nearly 56% of them. Some of the questions asked in the sur-

vey are listed below:

- Rate the difficulty level for each of the tasks category
- How much impact phone screen size had on your performance for different categories of tasks?
- How much impact phone make and model (i.e. quality of phone) had on your performance for different categories of tasks?
- Where were you mostly while performing tasks?
- In future, on which devices would you prefer to do such tasks? (Desktop/Laptops or Smartphone/Tablets)
- Any other inputs to improve Wallah platform?

We tried to get insight into the user-perceived difficulty of solving various categories of tasks as shown in Figure 9. The Human OCR Task and Human OCR Validation task were perceived as the easiest ones with 87% and 94% survey participants rating their difficulty levels as "Very Easy" or "Easy" respectively. More than half of the survey participants rated the Image Tagging Task and Audio transcription tasks with "Medium" difficulty. Only 6% of the survey participants found translation tasks as "Hard" and none of them rated translation validation Task as "Hard" or "Very Hard". The Video Tagging Task and Video Tagging Validation Tasks were perceived as the most difficult ones with nearly 60% and 40% of the survey participants rating their difficulty level as "Hard" or "Very Hard" respectively. When the users were asked about the impact that screen size had on their task performance it was found that most of the survey participants think that screen size had little or no Impact while performing tasks. Exceptions were Image Tagging and Video Tagging Tasks with nearly 64% and 52% survey participants feeling that the screen size had quite a lot of impact on performance. Similar statistics were observed for impact due to the phones' make and model on task performance.

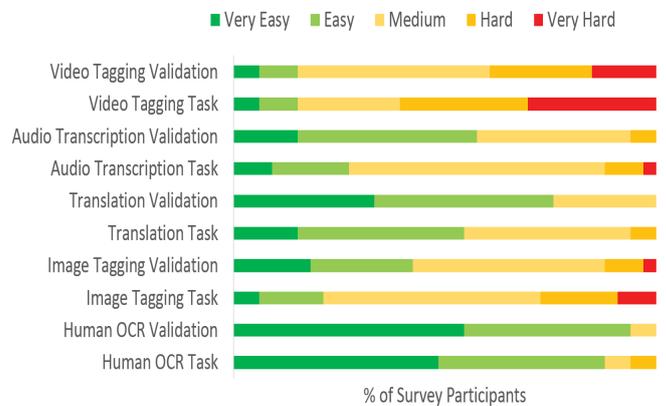


Figure 9: User perceived difficulty level of different task categories

We also wanted to know where people prefer to solve tasks, when asked for, it was observed that people equally preferred Home as well as Workplace/Institute to solve tasks. 45% of the survey participants said that they solved most of the tasks while at home and 48% of them while at Workplace/Institute. 6% of the survey participants said that they preferred to do tasks while traveling on a bus or train. On being asked about the time preference to solve such tasks nearly 90% of the survey participants selected evening and night.

Nearly 50% of the participants said that they did not face any bandwidth issues while fetching and solving tasks, 36% faced bandwidth issues, and the rest did not remember. We wanted to know the opinion of the participants about mobile being the key-player in crowdsourcing domain. When asked on which devices the participants will prefer to do such tasks, nearly 67% of participants chose Smartphone/Tablet over Desktop/Laptop. The participants also suggested a number of improvements for the platform, most of which were UI enhancements that we are examining and will try to incorporate in the forthcoming versions of the platform. Some of the participants also suggested that we should include gamification and leader boards to make task solving more competitive. Some users also expressed their satisfaction while using the platform "The GUI is simple and efficient. It clearly displays what the user needs to do." The participants ranked the platform usability at 5.3 on a Likert scale of 1 (Not usable at all) to 7 (Very Usable) and showed their willingness to use it regularly.

5. DISCUSSION

This paper presents Wallah, a task-centric mobile-based crowdsourcing platform designed especially for developing countries like India. The Wallah platform supports task-centric applications to minimize the impact of screen size and performs caching of crowdsourcing tasks to deal with network limitations. We implemented the Wallah platform for Android, and designed some task-specific applications (human OCR, image tagging, language translation, audio transcription, and video tagging) with specialized interfaces. We had a successful deployment of the developed platform among 59 crowd-workers in Delhi and Bangalore for two weeks and received more than 16000 crowdsourcing contributions from crowd workers.

Our field deployment validated the fact that there is a large heterogeneity among devices. We found 18 different screen sizes and 48 different phone models in the pool of 59 crowd-workers. We observed that there are a few crowd-workers who are super-active on the platform and performed most of the work. We found varying task accuracy for different categories of tasks. However, we observed that task accuracy decreases whenever the complexity of the task increases. For instance, video tagging tasks were performed with relatively lower accuracy (i.e., 63%) compared to others. We evaluated the impact of screen size on task completion time and quality for different categories of tasks, and surprisingly found that a larger screen size does not improve task completion time or quality. We performed an evaluation of usage of different specialized interfaces and found that crowd-workers perceive that these interfaces helps in reducing the task completion time. Also, crowd-workers with small screen size are more likely to use these interfaces as compared to other for specific application such as Image Tagging.

We believe that there is still a lot of room for improvement in enabling complex tasks on mobile devices and that there needs to be more work done—especially in designing task-apps that increase task quality. We conducted a post study survey to get qualitative feedback from the users and received satisfactory responses from them regarding the usage of the platform. With our deployment, study, and survey we can conclude that a system like Wallah can be a strong bootstrapper in the crowdsourcing domain, especially for developing countries like India.

6. REFERENCES

- [1] YouTube Auto Captions Feature, <https://support.google.com/youtube/answer/3038280?hl=en>
- [2] Smartphone Users Growth, <http://www.dazeinfo.com/2014/03/20/smartphone-users-doubled-urban-india-recorded-89-yoy-growth-2013/>
- [3] PC vs. Mobile Internet Users, <http://www.reuters.com/article/2012/07/19/us-china-internet-idUSBRE86I0FC20120719>
- [4] Android OS Versions, https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net
- [5] China 3G Penetration, <https://in.news.yahoo.com/billion-mobile-phones-china-3g-penetration-low-042833846.html>
- [6] India 3G Penetration, <http://www.zdnet.com/in/3g-subscribers-form-2-percent-of-indias-mobile-users-7000004883/>
- [7] Open Cell ID Database, www.opencellid.org
- [8] Mp3splt Project, <http://mp3splt.sourceforge.net>
- [9] Rajan, Vaibhav, et al. "CrowdControl: An online learning approach for optimal task scheduling in a dynamic crowd platform.", In proceedings of ICML Workshop: Machine Learning Meets Crowdsourcing, Atlanta, Georgia, USA, 2013.
- [10] Mao, A., Kamar, E., Chen, Y., Horvitz, E., Schwamb, M. E., Lintott, C. J., & Smith, A. M. (2013, March). Volunteering Versus Work for Pay: Incentives and Tradeoffs in Crowdsourcing. In First AAAI Conference on Human Computation and Crowdsourcing.
- [11] A. Finnerty, P. Kucherbaev, S. Tranquillini, and G. Convertino. Keep it simple: Reward and task design in crowdsourcing. In Proc. of Italian CHI, 2013.
- [12] Raptis, Dimitrios, et al. "Does size matter?: investigating the impact of mobile phone screen size on users' perceived usability, effectiveness and efficiency." ACM Mobile HCI '13.
- [13] Jassal, Prateek, et al. "Unity: Collaborative downloading content using co-located socially connected peers." In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013.
- [14] N. Eagle. txteagle: Mobile crowdsourcing. In Proceedings of the 3rd International Conference on Internationalization, Design and Global Development: Held as Part of HCI International 2009, IDGD '09.
- [15] P. Narula, P. Gutheim, D. Rolnitzky, A. Kulkarni, and B. Hartmann. MobileWorks: A mobile crowdsourcing platform for workers at the bottom of the pyramid. Proc. HCOMP '11.
- [16] A. Gupta, W. Thies, E. Cutrell, and R. Balakrishnan. mClerk: enabling mobile crowdsourcing in developing regions. In Proceedings of the SIGCHI '12.
- [17] T. Yan, M. Marzilli, et al. mCrowd: a platform for mobile crowdsourcing. ACM SenSys'09.
- [18] Khanna, Shashank, et al. "Evaluating and improving the usability of Mechanical Turk for low-income workers in India." ACM DEV 2010.
- [19] V. Della, et al. Crowdsourcing to Mobile Users: A Study of the Role of Platforms and Tasks. DBCrowd 2013: First VLDB Workshop on Databases and Crowdsourcing.
- [20] M. Musthag and D. Ganesan. Labor dynamics in a mobile micro-task market. CHI, pages 641-650. ACM, 2013.
- [21] Yadav, K., et al. "Low Energy and Sufficiently Accurate Localization for Non-Smartphones." IEEE MDM'12.
- [22] K. Yadav, V. Naik, A. Kumar, and P. Jassal. PlaceMap: Discovering Human Places of Interest Using Low-Energy Location Interfaces on Mobile Phones. Accepted at ACM DEV 2014.