Mobile Services for Enhancing Human Crowdsourcing with Computing Elements

Julian Jarrett[†⊥], Iman Saleh[†], M. Brian Blake[†], Sean Thorpe[⊥], Tyrone Grandison[ǀ], Rohan Malcolm[⊥]

[†]Department of Computer Science
University of Miami
Coral Gables, Miami, Florida

[⊥]Computational Sciences Research Group
School of Computing and Information Technology
University of Technology, Jamaica,
Kingston, Jamaica

[ǀ]Proficiency Labs,
Ashland, Oregon, USA

{j.jarrett, iman, m.brian.blake}@miami.edu, sthorpe@utech.edu.jm, tgrandison@proficiencylabs.com, rmalcolm92@gmail.com

Abstract — **Crowdsourcing enables one to leverage the power of the crowd. Normally, it involves utilizing humans for tasks that machines have difficulty performing. We propose a system, delivered as a mobile service, which dynamically adapts to the application domain and selects a combination of human and machine crowdsourcing components. Our work is towards the design of elastic systems that adaptively optimizes the use of human and automated software resources in order to maximize overall performance. We propose a performance model that predicts both human and machine outcomes for a certain task and then optimizes task assignment accordingly. Our experimentation shows that our proposed system significantly enhances the outcome precision of a crowdsourced task.**

Keywords: Mobile, Crowdsourcing, Services

I. INTRODUCTION

Crowdsourcing is a technique used to harness massive human resources to complete tasks [1]. With the emergence of smart mobile devices, potential human stakeholders are persistently connected to the Internet. This environment creates an opportunity to extend the computational reach of human expertise through their personal devices. In this paper, we propose *Elastic Mobile*, a *work-in-progress,* services-based system built on an elasticity framework, i.e. technique and implementation that leverages human and machine resources simultaneously in order to perform tasks.

Innovation within this paper is three-fold; an unique crowdsourcing architecture composed of specialized mobile web services, a new adaptive technique for combining effectively human and software efforts, and an implementation that supports mobile digital investigations for face recognition tasks. The choice of face recognition as a domain for our case study allows us to use social network users with mobile phones as *Human Computing Elements (HCEs)* to perform large-scale face recognition investigations. We show, through experimentation, that our framework optimizes the use of crowds and consequently increases the overall recognition performance.

II. RELATED WORK

Face recognition has multiple applications in forensics [2], such as suspect tracking and post event analysis. Despite numerous developments in the area of face recognition techniques, lighting, image quality, matching criteria and other aspects challenge accuracy [2]. Leveraging crowds for this domain is an obvious next step.

In addition to the authors' overview of using social networks for crowdsourcing [3], Gao et al. [4] highlighted the benefits of using crowdsourcing and social media to rapidly collate data for disaster relief management systems. In the aftermath of disasters, many posts to social media platforms were made via mobile devices.

Lin et al [5] proposed AgentHunt, a probabilistic model wrapped in a partially-observable Markov decision process to switch between workflows for crowdsourcing on Amazon Mechanical Turk. They assert that task designers experiment with multiple workflows to ascertain one that produces the best performance outcomes. Workflows may differ slightly in their presentation of questions, graphics or user interfaces to the human workers. One of these workflows is typically selected to go into production, based on results from earlier testing. The authors argue that selecting one workflow is still not leveraging the full potential of crowdsourcing and show with their method that alternative workflows can harmoniously attain higher quality results than selecting one workflow.

Yan et al [1] proposed MCrowd, an iPhone application that connects to major crowdsourcing platforms, e.g. Amazon Mechanical Turk and ChaCha, via a custom-designed MCrowd proxy server. This system supports two models of crowdsourcing; first the *requester model* that allows users (requesters) to posts tasks to the crowd. Second is the *worker model* where other users (workers) take up tasks that have already been posted by requesters.

Elastic Mobile utilizes the worker model, where autonomous machine components are combined adaptively with the expertise of humans via their smart, mobile devices.

Our work differs from other related projects in the fact that we focus on situations where the task for humans and for machine elements are *exactly the same*. Related projects focus on a variety of tasks for humans and machine elements where humans and machines address distinct tasks, distinct parts of tasks, or humans only address the tasks. Our research simultaneously evaluates where work can be split between the human and machine elements where both distribution of work and sequence of work can be used as a means for optimization. Our framework has a more distinct potential than related work to repeatably characterize the varying task effectiveness of humans and machines for large classes of application domains. As such, broader generalizations, based on the specific domain, are possible.

III. ELASTIC FRAMEWORK ARCHITECTURE

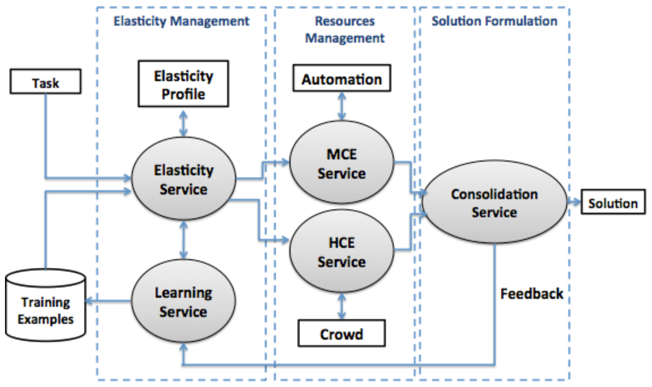

Figure 1. Proposed Elasticity Framework

We propose a service-oriented elasticity framework that leverages machine and human resources to efficiently and effectively perform a task. Services have been used extensively in modularization of workflow systems [6] and for appropriate scaling for systems with the demand for high performance [7]. Our proposed elasticity framework (Figure. 1) currently consists of three core components, an *elasticity management module*, a *resource management module* and a *solution formulation module*. Tasks and reference benchmarks for performance are primary inputs into the system via the *elasticity service*. The elasticity service determines the complexity of the task and the resources required to accomplish the task, whilst considering available system resources. Based on its analysis, the elastic service passes the task to *Machine Computing Elements (MCE's)* or *Human Computing Elements (HCE's)* or both. *Elastic Computing Elements (ECEs)* define when MCE and HCE effort is combined in the *consolidation service*. The consolidation service engages in a feedback loop that propagates results to the *learning service* where benchmarks are enhanced for future tasks. The services and interactions of our proposed framework are detailed below:

- *Elasticity Service*: This service orchestrates the use of MCEs and HCEs. It uses an *Elasticity Profile* (EP) to decide on the best combination of HCEs and MCEs that optimizes performance. The EP is domain-specific and captures the probability of successful task completion based on task modeling, as will be detailed later.
- *MCE Service:* This service manages the automated execution of a task.
- *HCE Service*: This service manages the execution of a task by a crowd of people. Different models of collaboration can be applied by customizing the HCE Service. For example, tasks can be done either by volunteers, or in return of compensation. The service can also acquire resources by employing online crowdsourcing systems, such as Mechanical Turk.
- *Consolidation Service*: This service consolidates the output of the HCE and MCE services. The service can decide, for example, on ignoring HCE output if certain quality threshold is not met. The service sends feedback to enhance future decisions.
- *Learning Service*: This service builds intelligence into the HCEs and MCEs orchestration process by building a knowledge base, denoted as *Training Examples* in the figure. This knowledge base is enriched by feedback from the consolidation service and is used to enhance the system performance based on historical data.

Our framework can be applied to different domains by plugging in the EP component of the corresponding tasks. The EP is defined as the weighted sum of a set of elasticity attributes:

$$EP = W_1 \times A_1 + W_2 \times A_2 + \ldots + W_n \times A_n \qquad (1)$$

where $A_1, A_2, \ldots A_n$ are domain-specific attributes that captures the task complexity and, $W_1, W_2, \ldots W_n$ are weights assigned to the different attributes. These weights can be initially set by experts, but then later tuned, by the learning service, based on the achieved performance. The EP enables optimal leverage of MCEs and HCEs resources. We denote the combination of MCEs and HCEs, to perform the same task, as ECEs or *Elastic Computing Elements*. The EP determines the level of elasticity, i.e. the percentage of human and machine time/resources, used by an ECE. For example, in case of face recognition, the EP's attributes capture the pictures' characteristics that affect face recognition. These characteristics include the pictures' size, color scheme, quality…etc. The EP is used to predict the precision of face recognition by both HCEs and MCEs. The purpose of our preliminary experimentation in this paper is to determine the nature of weights and how they vary in a real-life implementation.

Listing 1 shows the pseudo code of the Elasticity and Consolidation services. The Elasticity Service is assumed to have a queue of tasks. For each task, the service calculates the optimal HCE and MCE subtasks, based on the task's EP. The service sends the assignment to the HCE and MCE services. The services produce the corresponding solutions that are sent to the Consolidation Service. The Consolidation Service uses the EP to evaluate the output of the two services and generate an optimal solution.

```
Elasticity Service
Start
        Foreach Task T in the queue
          EP = Calculate_EP(T)
        //Based on EP, decompose T into substasks
        //to be executed by HCEs and MCEs
          {$T_{HCE}$,$T_{MCE}$}= Decompose (T, EP)
          Output {$T_{HCE}$,$T_{MCE}$}
      EndFor
Stop
```

```
Consolidation Service
Start
    Foreach Task T in the queue
        //Get Task solution from HCE and MCE Services
        {$S_{HCE}$, $S_{MCE}$}= getSolution()
        //Based on EP, calculate the optimal result
        $S_{ECE}$ = Optimize ($S_{HCE}$, $S_{MCE}$, EP)
        Output $S_{ECE}$
    EndFor
Stop
```

Listing 1. Pseudo-Code of Core Services in the Proposed Elasticity Framework

A. *Case Study: Mobile-Based Face Recognition*

We apply our proposed framework to the problem of face recognition. We propose the face recognition architecture shown in Figure 2. The architecture uses our elasticity framework, implemented by the elasticity server shown in the figure, to classify unrecognized pictures based on a set of reference pictures. The software works by recognizing faces from the reference images and building a database that connects faces to names. Given, a new unrecognized picture, the software normalizes it by transforming it into a gray-scaled picture that is sized to standard dimensions. It then applies bit values comparisons, with bit values in the range 0-255, to calculate a matching percentage. We set a matching threshold such that pictures with matching percentage more than 60% are marked as potential recognized faces. Typically, one or more reference picture is matched to each unknown picture. We use the number of positively matched pictures (P) of all suggestions (S) to characterize the systems' EP (P/S) as it's an indication of the level of uncertainty. Pictures with high uncertainty are sent to the crowd for recognition via an Android mobile application. The pseudo code of the face recognition system is shown in Listing 2.

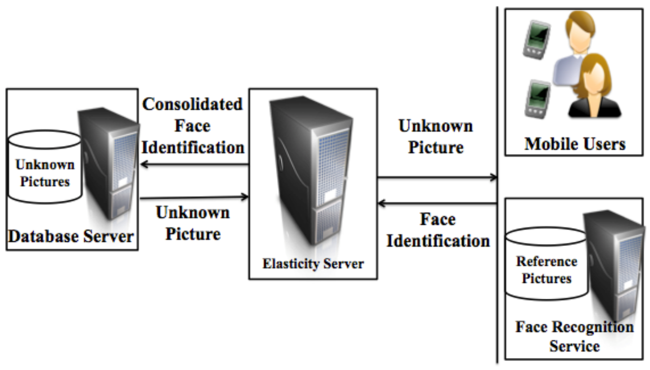

Figure 2. Physical Setup for Mobile Face Recognition

```
Procedure WorkFlowECE
Start
        Foreach ReferencePicture to Add to System
                Input ReferencePicture in ReferenceSet
        EndFor
        Foreach TestingPicture to Add to System
                Input TestingPicture in TestingSet
        EndFor
        Foreach TestingPicture in TestingSet
                ListSuggestions=Call_
                MCEMatch(accepts_
                TestingPicture, ReferenceSet)
        EndFor

        Send ListSuggestions to HCE via CrowdSourcing

        //WorkFlow executed on Mobile Device to
        //consolidate MCE with HCE to give ECE
        Foreach MCESuggestion in ListSuggestions
                Submit HCE Feedback for _
                MCESuggestion to Learning Service
        EndFor
        //WorkFlow on Mobile Device Ends

        //Begin MCE vs ECE analysis in Learning
        //service
        EP-ECE = Analyze HCE Feedback for
        MCESuggestions  //EP=ECE is EP-MCE //+ EP-
        HCE
        EP-MCE = Analyze ListSuggestions for Positive
        Identification
        Results = Compare EP-ECE vs EP-MCE
        Show Results
Stop

Function MCEMatch Returns ListSuggestions Accepts_
 TestingPicture, ReferenceSet
Start
        Foreach ReferencePicture in ReferenceSet
                ResultSimilarityMatch=Compare _
                TestingPicture to ReferencePicture using_
                MCE Face Recognition with bit _
                Threshold value 50
                If ResultSimilarityMatch > 60%
                        Add to ListSuggestions
                End If
        EndFor
        return ListSuggestions
Stop
```

Listing 2. Showing Algorithm Psuedo-BASIC Workflow of Mobile Face Recognition System

Mobile users are presented with the unknown picture and the names of people of the potential matches generated by the software. They have the option to confirm one of the potential matches, confirm that the person is someone else or choose that they don't know the person in the unknown picture (Figure 3). As users select the options available to them, this information is sent to the server and persisted for future statistical analysis for supporting the learning service.

B. *Experimentation: Mobile Face Recognition*

In this study, we apply this technique to a social media digital forensics application. Ideally to allow admissibility of evidence in court, it is assumed that the certainty value of the data as submitted by the law enforcement team has the highest levels of precision and accuracy. The baseline evidence data contains images and facial data profiles of 23 popular persons as the training set. Varying pictures of the 23 individuals were gathered to build a test set. In initial experimentation, the test set was sent to the elasticity service, which optimized MCEs and HCEs to produce an ECE. The elasticity service had no prior performance history of the MCE and HCE components hence used both MCE and HCE for the initial run of the system.

IV. Human Computing Elements using Elastic Mobile

In our physical setup (Figure 2), the crowd consisting of multiple HCEs received the face recognition tasks through a custom Android mobile application, distributed through the HCE XML Web service component (Figure 1) physically located on the elasticity server (Figure 2). Users were provided the testing data set on their mobile devices and asked to verify the identification of the person in the picture based on preliminary suggestions from the MCE component, using their own knowledge or stating that they did not know the individual in the picture. Crowd feedback was collected by the HCE web service and sent to consolidation service.

V. Results

Figure 3 shows the results from the 30 respondents. In 9 of 23 test cases, HCE's were able to confirm the individual from the testing set, based on MCE suggestions. For all test cases, HCE's identified the individual from the testing set 40.64% of the time using MCE suggestions provided the suggestions included the correct individual, 47.64% of the time from their own knowledge and didn't recognize the individual for 41.4% of the occasions. Figure 3 details the 30 respondents participating in a mobile face recogniton experiment. HCE certainty (Figure 4) revealed HCE's having an average certainty of 69.13% with maximum certainty of 100% a minimum of 16.67% and a range of 83.33%.

VI. Conclusion and Future Work

The results from our case study show that HCEs provide an average level of certainty at 69.13%. We propose that Elastic Mobile be used to increase the outcome quality of crowd-sourced tasks. In the future, we will use crowd-sourced feedback from a much larger sample set to enhance predictions of the MCEs through the learning service in our framework (Figure 1), especially in situations where MCEs produced no suggestions. In addition to simple, existing feedback options, we will extend our mobile face recognition application to allow HCE's in the crowd to submit other forms of data to the elasticity server for interpretative forensic analysis and crime scene reconstruction as new formal problems of concern that motivates the need for the work so far presented in this study.

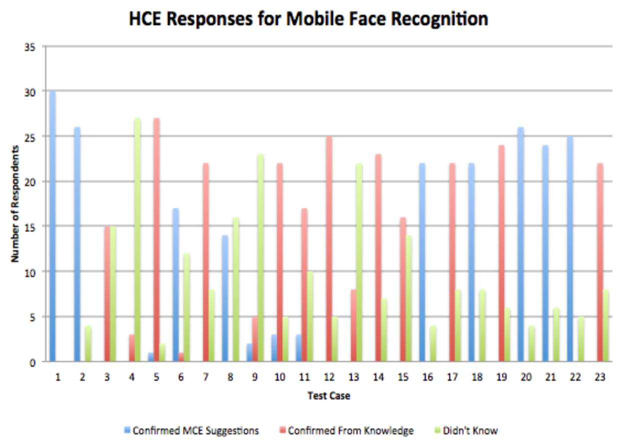

Figure 3. Responses for Mobile Face Recognition

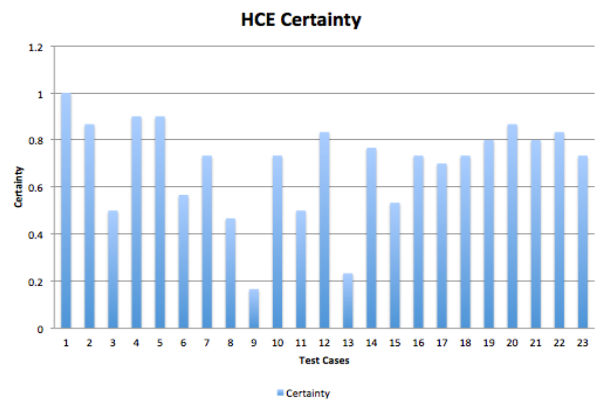

Figure 4. HCE Certainty for Mobile Face Recognition

References

[1] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner. "mCrowd: a platform for mobile crowdsourcing." In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 347-348. ACM, 2009.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. "Face recognition: A literature survey." *Acm Computing Surveys (CSUR)* 35, no. 4 (2003): 399-458.

[3] W. Tan, M. B. Blake, I. Saleh, and S. Dustdar. "Social-Network-Sourced Big Data Analytics", *IEEE Internet Computing*, Vol. 17, No. 5, pp. 62-69, Sept/Oct 2013

[4] H. Gao, G. Barbier, and R. Goolsby. "Harnessing the crowdsourcing power of social media for disaster relief." *IEEE Intelligent Systems* 26, no. 3 (2011): 10-14.

[5] C. H. Lin, and D. S. Weld. "Dynamically switching between synergistic workflows for crowdsourcing." In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

[6] M. B. Blake, and H. Gomaa. 2005. "Agent-oriented compositional approaches to services-based cross-organizational workflow." *Decision Support Systems* 40, no. 1 (2005): 31-50.

[7] Y. Wei and M. B. Blake. 2010. "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities." *IEEE Internet Computing 14, no. 6 (2010).*