Weed Infestation Identification Using Hierarchical Crowdsourcing

Mahbubur Rahman*, Brenna Blackwell**, Nilanjan Banerjee*, Dharmendra Saraswat+

Computer Science and Electrical Engineering, University of Maryland, Baltimore County*, Computer Science and Computer Engineering, University of Arkansas**, Fayetteville, Biological and Agricultural Engineering, University of Arkansas, Fayetteville+

{mrx056, bab011}@uark.edu, dsaraswat@uaex.edu, nilanb@umbc.edu

Abstract

Weed infestation is a common problem in agriculture that adversely affects crop production. Given severe constraints on the budget of many land-grant universities due to the economic downturn, outreach services have taken a hit. To adapt to current economic climate without adversely affecting the quality of outreach program for weed management, we present a hierarchical system that uses image captured by smartphone, a backend image processing algorithm, and two levels of crowdsourcing approaches to identify weed images. The first of the two crowdsourcing levels consist of non-expert crowd contributed by Amazon Mechanical Turk and the second level consisting of expert crowd comprising of county extension agents. A probabilistic decision engine was used, in an unsupervised manner, to determine the suitability of two levels of crowdsourcing approaches for identifying the weed image. The designed system was found to have low latency with high accuracy for identifying weed from captured images. The designed system accurately identified test weed within 3 hours of its submission using minimal human intervention. The system has shown good potential to support weed management related outreach programs in land-grant universities.

Keywords: Weed identification, Crowdsourcing, image, processing, Smartphones
1. Introduction

Weeds compete with crops for light, water, and nutrients. If left uncontrolled, weeds will significantly reduce a farmer’s yield. Methods of weed control and management are constantly being updated, and more efficient practices emerge as a result of study and observation, but farmers can only benefit from these advances if they have access to the most current information. The present extension approach to disseminate control practices for weed infestations relies on traditional techniques such as extension publications, county meetings, and one-on-one consultations. The economic downturn has adversely affected outreach budgets of several land-grant universities. This has led to concerns that universities could scale back their outreach mission of supporting communities in the time of need [1]. Several states have been restructuring their extension infrastructure to cope with the harsh economic realities. Given this reality, it is time to start thinking of some innovative approaches for keeping outreach services viable and useful. We have used weed management program as a test case to propose a novel diagnostic and control system that could be used as a back-up system to the existing programs. A typical weed identification call from a producer results in field visit by a county extension agent. The agent either completes the diagnosis or refer to the specialist who in turn provide the diagnostics and refer control practices.

Smartphones have penetrated the rural population both in developing and developed countries. For instance, low-end Android phones that provide basic cellular data plans, a camera, and location information are common devices for farmers and extension agents. In a recent initiative in Arkansas, for example, the extension agents were provided with data plan-enabled iPads. We leverage this observation to design a system that can provide low latency, high accuracy, and low cost control practice dissemination to farmers when their crops are weed infested. At the core of our technique is a weed identification system that leverages the concept of crowdsourcing [2, 3] —using a human network to solve a computationally hard problem— and functions as a weed management information distribution system. The initiator of the crowdsourcing effort would be the outreach arm of land-grant universities and the crowd would consist of non-experts provided by services such as Amazon Mechanical Turk and experts employed by land-grant universities. Such a system would benefit the farmers of the state who are in need of expert advice.
The overall architecture of our proposed system is illustrated in Figure 1. We use a combination of hierarchical crowdsourcing and image analysis to provide timely and accurate feedback to farmers requesting information on plausible weed infestation in their crop fields. The end-user, for example a farmer, is assumed to own a camera, and GPS-enabled smartphone. When the farmer determines that his crop is infested by weeds, he takes a picture using the smartphone camera. Our designed client application uses the picture, automatically geo-tags it with GPS coordinates, inserts any comments that the farmer may have, and uploads it to our backend database using a custom designed webservice. The webservice first performs an image analysis that compares the picture of the weed with a database of weeds seen in the region in the past. The image analysis ranks the images in the database based on the overlap with the weed. The image processing acts as a low latency mechanism to determine which weed has infested the farmer’s field, and also acts as a filter to narrow down a subset of plausible weeds that might be similar to the weed infecting the crop field. The webservice then uses two levels of crowdsourcing to identify the weed. First, it uses a set of non-experts, from a portal like Amazon Mechanical Turk to extract crude but low cost feedback on what the weed might be. If the non-experts cannot accurately determine the weed, the system consults the experts (extension agents). A decision engine determines whether the system is confident enough on what the weed might be. The decision engine is applied after the image analysis and non-expert recommendation stage to determine if the next step is required. This helps minimize the use of the extension agents and reduces the overall cost.
and latency associated with the weed identification process. Once the system identifies the weed, it uses a database of control practices to provide accurate and up-to-date suggestions on how to mitigate the weed infestation.

The design, implementation, and evaluation of our weed management architecture presents three novel research contributions. Our first contribution is a system that combines image analysis (machine intelligence) and crowdsourcing (human intelligence) to accurately infer the type of weed infecting a farmer’s field. The technique uses a probabilistic decision engine to provide high accuracy, low latency and low cost inferencing. Our second contribution is the use of two levels of crowdsourcing, non-experts from services such as Amazon Mechanical Turk [4] and experts like extension agents. We propose another probabilistic technique to determine when the system is confident of its inference, and whether the second layer of crowdsourcing from experts is required. This helps minimize the use of experts, and can consequently reduce the cost of inferencing.

Our third contribution is an end-to-end system that includes smartphone applications for farmers and experts, and backend services that house the image processing and crowdsourcing logic. The present article evaluates the end to end system’s latency, accuracy, and energy consumption characteristics. The practical deployment of the system shall be covered in future articles.

2. Related Work

Identifying weeds plays a major role in controlling infestations through timely use of herbicides in the crop field. If a farmer receives timely and detailed information on control practices for weeds, he would be able to more efficiently spray herbicide in his field, reducing environmental contamination and damage to crops. The smartphone-based automated architecture developed by us builds on previous research on image analysis for weed identification and application of crowdsourcing techniques in unrelated application domains.

2.1. Image Analysis

Several techniques for identifying weeds using contextual data have been proposed. To augment automated image analysis, for instance, researchers
have used contextual data on lighting conditions [5], color and shape [6], soil conditions, weed’s age, and information on the specificity of the crop field [7]. Many of the proposed weed identification processes in crop fields use machine vision [8, 9, 10] systems, but the morphological and texture parameters used in image analysis [8, 11, 12] can be complicated and computationally expensive [13]. These also need to run on dedicated desktop PCs to meet the required computational needs.

In addition to using contextual data, research has been performed in the field of weed sensing in order to discriminate between weeds and plants for specific species like tomato seedlings [14], using shape features of crops [15, 16, 17, 18, 9, 19], spectral analysis [20, 18], fractal analysis of leaf shapes [21, 22] using fourier transforms, hadamard transformation and wavelet [23], haugh transformation to find weeds between two rows of crops, and texture of images from a canopy [22, 18, 10]. However, most of these techniques have not been deployed in actual farms. As we show in our evaluation, image analysis alone is not reliable enough to detect weeds accurately.

2.2. Crowdsourcing Techniques

Although computer vision and image analysis are powerful techniques for identifying weeds and crops, they have several limitations and cannot guarantee identifying weeds accurately. For instance it is difficult for automated algorithms to identify weeds at different stages of growth, or when using images taken at different angles and lighting conditions. Moreover, it is impossible to identify weed species that have mutated or are new. Image processing using a human crowds of experts and non-experts has been shown to be more powerful [24] than computer vision for several application. Popularly termed as crowdsourcing, a network of humans can be used to solve challenging and computationally expensive problems that machine intelligence cannot accurately solve. The pioneer work of Luis Von Ahn on reCaptcha [25], for instance, uses humans to digitize old books that augments optical character recognition algorithms. The EPS games [26] also use humans for determining descriptive labels for images. Image searches using crowdsourcing can be tuned to have high accuracy and low cost [27].

3. Design Goals

Our goal for the weed identification system was to provide a low cost, low latency, accurate, and highly usable system to automate the weed detection
and mitigation problem. We strictly adhered to the following design goals while implementing our system:

- **Reduce the weed identification burden for extension agents:** In an effort to minimize cost, we proposed that a hierarchical architecture where submitted requests are first analyzed using low-cost methods. These methods include automated image processing as well as crowd-sourcing, by way or using non-experts from Amazon Mechanical Turk (AMT). A farmer’s request would then be sent to an expert extension agent only if the system cannot identify the weed using the prior methods.

- **Provide low latency and high accuracy weed identification:** The goal of the proposed hierarchical architecture was to create high accuracy, low latency weed identification. We postulated that if the system were set up such that a farmer’s request is sent to an expert without any intermediate filters, the experts may potentially receive too many requests. Due to the volume of requests they would receive and that county extension agents have further obligations than weed identification, the experts might not be able to respond to them all while at the same time. This would inevitably lead to long response times. Moreover, there is no underlying mechanism to determine if the responses provided by the lower levels of the hierarchy are accurate. To mitigate the problem, we propose a decision module that measures the accuracy of the response at all levels of the system hierarchy. This decision module is intended to reduce the number of requests forwarded to the extension agents, reducing their weed identification burden and freeing them up to complete their other obligations.

- **Easy to use end-user applications:** The adoption of the weed identification system is predicated on creating intuitive smartphone interfaces for both the farmer and the expert. The user interface is designed such that data is presented to both farmers and experts in a clearly structured way, utilizing both table and map views. Wherever necessary, instructions have been incorporated to guide the user in the operation of various features.

while incorporating expert feedback from extension agents and crop boards. Data is presented to the users, both farmers and experts, in a
clearly structured way utilizing both table and map views. When user input is required, the interface is designed with instructions that are provided if the user needs further guidance.

4. System Architecture

Our weed identification system architecture is illustrated in Figure 1. As described in §1, the system uses a combination of hierarchical crowdsourcing based on Amazon Mechanical Turk and experts, augmented with automated image processing, to accurately identify wees and provide control practice recommendations to farmers. Interacting with this backend are the client-side mobile applications. We next describe our smartphone applications and the backend image processing logic.

4.1. Smartphone Applications

The system is comprised of two smartphone applications. The first smartphone application resides on a farmer’s smartphone. The farmer can use the application to take geo-tagged images of weeds in his field. These images, annotated with text and audio comments, are transferred to our backend server over a cellular or Wi-Fi connection. The second smartphone application resides on the expert’s device. If our backend logic cannot reliably infer the weed species using image processing and the Amazon Mechanical Turk crowd, it notifies a set of select experts, soliciting input on the type of weed. The smartphone applications (see §6 for screenshots) provide features allowing the users to keep track of the requests submitted by the farmers, responses from the system and experts, and weed control practices. The app also features a weed database that the users can manually browse and compare weed images against.

4.2. Backend Image Processing Logic

The request query from the farmer’s smartphone, which consists of a geo-tagged image and comments, is sent to the backend server. The first step is to analyze the image using an automated image processing engine in order to identify the query weed image. While image processing can uniquely identify images if reliable image features are available, it fails to correctly identify images in several cases. These cases are such as when the images are taken when the camera lens is mis-oriented, or when images are taken under poor
Figure 2: The figure illustrates the two phases of the image processing algorithm. During Phase 1, the features are extracted, clusters are created for each feature, a centroid for each cluster is calculated, and the centroids are indexed in a database. During Phase 2, the same features and cluster centroids are calculated for the query image, and the distance is calculated with the centroids in the indexed database. The top five images are calculated for each figure, and a majority vote is used to find the five images that closely match the query image.

lighting conditions. Since farmers are not expert photographers, and off-the-shelf cell phone cameras are used to take the images, it is imperative to assume that the image quality may be suboptimal. Additionally, because our image processing scheme compares the query image with a database of weed images, it is impossible to identify unknown invasive weed species or a new breed of weeds [28]. While the image processing engine may not be able to identify the weed in several cases, it can, however, narrow down the search to a smaller subset of candidate images. Narrowing down the image search to a few candidate images is especially important because our system depends on a human crowd to identify the correct weed image. An Amazon Mechanical Turk worker, for instance, is unlikely to browse through hundreds of images in our weed database to identify the correct weed image. In fact, research in the past has shown that response from human crowds are more accurate if the number of choices provided to the end user is small [27].

Our image processing engine, therefore, takes a preset number of images as input \((n)\), and outputs the top \(n\) images that are a close match to the weed image provided by the farmer. We set \(n = 5\) for our implementation. The image processing algorithm uses a concept of multi-feature fusion to extract
the best candidate set of five images that match the original weed image. The multi-feature fusion algorithm is illustrated in Figure 2. The algorithm works in two phases. During an offline training phase (Phase 1), our algorithm extracts multiple features from the database of images. Specifically, our system uses the following features: sift-scale invariant features, cedd-color edge directory descriptor, pixel correlograms, fcth-fuzzy color, and texture histogram [29]. We have chosen these features as they encapsulate a wide range of characteristics of weed images, and individually can capture leaf shapes, background color, plant texture, and edges characterized by plant stems. We use an open source lucene indexed-based image processing library called LIRe to build our feature extraction databases. The feature values for the images are clustered and the centroid of the cluster is stored as a byte payload in a lucene-based index [30]. For instance, if the color histogram is a feature extracted by our system, three clusters are created for the Red (R), Green (G), and Blue (B) values for each image, and the centroid of each cluster is stored in the index database. During phase 2 of the algorithm (online phase) the query image from a farmer’s smartphone application is processed and the same set of features as Phase 1 are extracted. The feature values are clustered and the center is calculated, like in Phase 1. For each feature \( f \), the top five images \( I_f = \{I_1, ..., I_5\} \) are selected with shortest distance between the center of cluster for feature \( f \) and the center of the same feature for the query image. To determine the five images closest to the query image, the system calculates a rank for each image in the database, \( R_i \), which is equal to the number of sets \( I_f \) that it appears in. Then, the image processing algorithm chooses the top five images with the highest rank. For each image, the image processing algorithm also outputs a probability, \( P_i \), which is the probability that the image is the same as the query image. These five images are used as input to our novel hierarchical crowdsourcing algorithm that combines Amazon Mechanical Turk and experts to determine the weed that closely matches the query image, or registers a new weed species.

5. Hierarchical Crowdsourcing for Weed Identification

Humans can perform tasks using their vision and thinking capabilities, a process that remains very difficult or sometimes impossible for computers to replicate [31]. Digital image identification in our application setting, for instance, is a problem due to the nature of the pictures being taken. The
image processing algorithm described above for example, can suffer from false positives if the image is not present in the weed database, such as in the case of invasive weed species. In our system, therefore, we augment our image processing algorithm with a crowdsourcing infrastructure that solicits human input to identify images.

Weed identification for humans, however, can also be a fairly involved process. To understand why, consider the following example: The weed image taken by the farmer could correspond to a weed at different stages of growth or images taken at different angles (Figure 3 shows an example). At different stages of growth, weeds can look different. To a non-expert human eye, it is difficult, if not impossible to infer that the images belong to the same weed. Experts in the area of weed identification are extension agents—individuals (described in §1) who manage and monitor large agricultural farms at the county level. Unfortunately, the number of extension agents in a state are very few, hence expertise is a very scarce and expensive resource, and asking extension agents to identify weeds can lead to long latencies and can be expensive in terms of the dollar cost of engaging these agents.
Therefore, to minimize this cost and improve latency performance, we propose the above hierarchical approach. The first layer is the image processing subsystem that was discussed in §4.2. The image processing algorithms act as a filter that narrows down the image search to a small and tractable number of candidate images. The second layer uses an untrained human crowd, provided by popular web services like Amazon Mechanical Turk (AMT) to identify the weed. The AMT crowd is a low cost, low latency image identification subsystem. We have designed a decision engine that automatically infers if the AMT crowd has correctly identified the query weed. If the decision engine determines that the system is not confident that the correct weed has been identified, we resort to soliciting input from extension agents. Extension agents install the expert version of our smartphone application and are asked to rank the five images based on how close the image is to the query weed. The extension agents can consult the entire database of images to identify the correct weed image. The inputs from different extension agents (if available) are then merged using majority vote, and the best weed image and the corresponding control practice is disseminated to the farmer. In the next subsections, we discuss the use of AMT, expert extension agents, and our underlying probabilistic decision engine.

5.1. Low Cost Non-expert Crowdsourcing

Amazon Mechanical Turk [32] (AMT) is an Internet marketplace for human crowdsourcing. This marketplace enables a computer system, known as Requesters, to coordinate the use of human intelligence and machine intelligence to solve a problem that would be difficult or impossible to perform using only machine intelligence. The Requesters can publish tasks known, as HITs (Human Intelligence Tasks) [33], such as choosing the best images, writing product descriptions, run online surveys or identify performers on music CDs. Workers who are called Providers in Mechanical Turk’s Terms of Service, or, more colloquially, Turkers [32]. These workers can browse existing tasks for which they are eligible and complete them for a monetary payment set by the Requester. The monetary payment amount is preset by the Requestor and can vary depending on the task, but is always greater than 1c per task. The turk web services also provides tools to keep track of the workers and their performance. A Requestor can choose workers for the task based on past performance [32]. The time of publishing HITs, the time the HIT is accepted by a turker, and when a response was received can be tracked using the Mechanical Turk API. In our system, we construct a re-
quest using our query image, and the five candidate images that are outputs from the image processing algorithm, and publish the request to the AMT webservice. To publish the requests to AMT, we used the Java-based application interface that communicates with the Amazon Web Services. The workers are requested to respond with images similar to the submitted image. The detailed rules of answering the requests are also included to provide the workers with a better understanding of the questions before answering them. After publishing this request to the AMT, the system waits for the workers to accept the request and provide answers.

The expert human crowd comprises of individuals who have in-depth knowledge about the weeds. We have recruited these agents from the State of Arkansas. In the hierarchical identification system, experts play the role of a top hierarchical object. In many cases, both the image processing algorithm and the non-expert turkers may fail to identify the query weed. This may arise when the query weed image looks different from the weed image in the database or the weed image is non-existent in the weed database, such as in the case of an invasive species. Our probabilistic decision engine (described below) automatically infers if the system is sufficiently confident that the weed identified is the correct one. If that is not the case, the system solicits input from the experts in our database using the expert version of our smartphone application. It is assumed that the experts have apriori installed our smartphone application.

5.2. Probabilistic Decision Engine

Our weed identification system works with two constraints: (1) a constraint on the acceptable latency that the end-user (farmer) can tolerate; and (2) a constraint on the number of dollars that the system can invest in crowdsourcing results from a public webservice like AMT and the time offered by extension agents. Throughout our system design, we assume that our system would be maintained by a government entity like the state, or private groups like crop commodity boards. In our system configuration, the image processing subsystem has been designed to have a low latency signature and zero dollar cost. However, the subsystem can be potentially inaccurate. The expert crowd is assumed to have a substantial dollar cost with it when compared to using an non-expert crowd like AMT. As future work we plan to accurately model the cost of using experts to respond to image queries. Therefore, the experts are assumed to respond to queries within a set period of time. The cost of using AMT is configurable and the
requester can determine the amount of money he wants to spend. The goal of our decision engine, therefore, is to calculate when the system is confident enough that a given weed has been correctly identified while satisfying the latency and cost constraints.

Our probabilistic algorithm, motivated by greedy Q-Learning [34] is illustrated in Algorithm 1 (also illustrated in Figure 4). The algorithm takes as input the $n$ images $\{I_1, ..., I_n\}$ that is the output of the image processing algorithm, a constraint on the dollar amount ($D$), and a constraint on the latency ($T$). The latency corresponds to the actual latency constraint set by the system or the farmer, after subtracting the estimated constant latency of using extension agents and the image processing subsystem. The dollar $D$ is used to determine the number of HITs that are published to AMT. For our present implementation, we use a combination of 1c HITs and 5c HITs. The proportion of higher cost HITs can be tuned to match the latency constraint—we plan to pursue this problem as future work. Our probabilistic algorithm is executed every time a response is received from an Amazon Turker. To understand how the algorithm works, lets assume that our system receives the $X_i^{th}$ response from Turk worker $W_i$, and has already received $X_1, ..., X_{i-1}$ responses from takers. The system tracks the profile

Figure 4: The figure illustrates the our probabilistic decision engine that determines the probability that image $I$ is the correct match for the query image after $X$ responses. The probability is a product of the quality of the AMT worker, Probability that the image $I$ is the correct match given $X - 1$ responses, and the probability that the image $I$ is the correct match given the $X^{th}$ response.
of the quality of every Turker $W_i$ in the form of a fraction $Q(W_i)$, where $Q(W_i)$ is the fraction of correct responses $W_i$ has provided to our system in the past. Therefore, $Q(W_i) = 1$ represents a high quality Turker who has provided the correct answer 100% of the time. Using $Q(W_i)$, the system calculates a probability $P(I_i | \#R = X)$, for every image $I_i$, that represents the probability that $I_i$ is the correct weed image given the number of responses $\#R$ is $X$. The probability is calculated using Equation 1.

$$P(I_i | \#R = X) = Q(W) \cdot P(I_i | \#R = X - 1) \cdot P_{curr}(I_i)$$  

(1)

$P(I_i | \#R = X - 1)$ is the probability that $I_i$ is the correct image after receiving $X - 1$ responses, and $P_{curr}$ is the probability of $I_i$ is the correct image given the $X^{th}$ response. $P_{curr}$ is calculated using a greedy theory technique proposed by Sheng et al. \cite{35}, and is explained through the following example.

The Amazon turker can rank up to three images out of five images that are provided through the HIT. However, the turker also has the option to rank fewer than three images. For instance, the Turker can rank one image as the exact match, or three images as Rank 1, 2, and 3. For the first case, we assign exponentially reducing probability to the images — 0.5 probability to rank 1 image, and distribute the rest of 0.5 to the four images (0.12 to each image). For the second case, the rank 1 image is assigned 0.5, the rank 2 image is assigned 0.25, and the rank 3 image is assigned 0.125, and the rest is divided between the two remaining images. Using this method of probability assignment, no image is assigned a 0 probability. Using this assignment, $P(I_i | \#R = X)$ is forced to be a positive value. The initial value of these probabilities correspond to the output of the image processing subsystem (recall that the image processing algorithms outputs the probability $P_I$ for each of the five candidate images that is calculated based on the distance measure of the algorithm). Finally, the probabilities are normalized such that $\sum_{I_i} P(I_i) = 1$. If the probability for a particular image is greater than a threshold (set to 0.8 in our system, but is configurable), the system assumes that the correct image has been identified. If after the latency constraint ($T$) is exceeded, none of the images have a probability of greater than 0.8, the system solicits feedback from the extension agents.

6. System Implementation

Our smartphone application will be available on iOS and Android, and the web. It is written in Objective C and Java, respectively on the two
Figure 5: (a) The Figure illustrates the screenshots from the iOS smartphone application. (b) The Figure illustrates the screenshots from the Android smartphone application.

mobile platforms, while the web-based client is written using a PHP web-service frontend. The image processing algorithm is implemented using Java. The data is stored in a mysql backend database. Overall, we have written 8200 lines of iOS code, 28, 465 lines of Android code, and 3000 of backend webservice and processing code. We have implemented several optimizations including background download of images, and caching of data on the smartphone to improve the performance of the system. We profile the memory and energy signature of the smartphone applications and latency characteristics of our backend in §7. Figure 5 illustrates the screenshots of our smartphone applications.

7. System Evaluation

The goal of our hierarchical crowdsourcing system is to provide accurate and low latency weed identification at minimal cost. A complementary design goal of the system, therefore, is to reduce the burden on county extension agents and other weed experts in the identification process. Hence, our evaluation focuses on the following key questions.

- What is the weed identification accuracy of our image processing algorithm and the AMT crowdsourcing subsystem?
- What is the latency of weed identification for our hierarchical system?
What are the trade-offs between the cost of publishing AMT HITs and the latency of turker responses?

We also present micro-benchmarks on the energy and memory consumption overheads of our implemented smartphone applications.

7.1. Experimental setup

Here, we discuss the the experimental setup of our end to end system, as illustrated in §6, the image processing technique described in §4.2, our probabilistic decision engine. In the experimental setup, we break down the system into three components—image capture and upload using the smartphone applications, backend image analysis, and offload of images to Amazon Mechanical Turk—and rigorously evaluate each component. While we do not directly evaluate the accuracy and cost of using extension agents in the identification process, based on feedback from domain experts, we assume that both the cost and accuracy of extension agents’ identifying weed images is high. To evaluate the AMT component, we created 70 instances of each request. For each of the requests, the cost of each HIT was either 1c per HIT or 5c per HIT. The HITs are comprised of two components—the candidate image and the top five images that the image processing algorithm selected from our apriori collected weed image database. On average, we received 66 responses. For our evaluation, we use three metrics—accuracy of identifying a weed image, the average cost of identifying a weed image using AMT, and the latency of identifying a weed image.

7.2. System Accuracy

In our first set of experiments, we evaluate the accuracy of the weed identification process using our system. Specifically, we examine the accuracy of our image processing algorithm and crowdsourcing subsystem. For our experiments, we used a set of candidate weed images that look different from the images in the weed database. This difference in the images could occur due to different stages of weed growth, or because the images were taken at different angles, lens exposure, and ambient lighting conditions. In several instances, we added artificial noise to the weed images to create candidate images. For each candidate image, we execute our image processing algorithm described in §4.2. The image processing algorithm identifies the top five images and ranks them by the average distance from the candidate image. Figure 7 plots the distance measure for the top five candidates across
Figure 6: (a) The figure illustrates the cumulative probability calculated using our probabilistic algorithm for nine independent HITs published to Amazon Mechanical Turk. (b) The figure shows the average of the probabilities for the nine HITs. The error bars are calculated using standard deviation. The figure shows that 0.8 is a reasonable threshold for our application and the system requires at least 10 responses before making a reliable inference.

Figure 7: The figure shows the distances calculated by the image processing algorithm for the top five candidates.

9 distinct runs of the algorithm. From the experiment, we find that the image
processing algorithm identifies the correct image in 89% of the cases in the top five choices. However, there is a considerable amount of uncertainty associated with the image matching algorithm identifying the closest image to the candidate image. As shown in the figure, the difference in the distance between the top candidate image and second best match is not small. Hence, it is likely that the algorithm will get confused between two images and will not output the actual image as a top candidate. Image processing, therefore, is insufficient for identifying weed images—the human crowd is required to solve this fairly complicated weed identification problem.

We next evaluate the accuracy of using the non-expert Amazon Turk crowd to identify the candidate image. Our HITs published to AMT allows each turker to rank up to three out of five images that are an output from the image processing algorithm. As described in §5.2, we use greedy theory to assign the probability, $P_{curr}$ that weed image $I_i$ is the best candidate image after the $X_{th}$ response. Based on this probability, we calculate $P(I_i | #R = X)$, the probability that image $I_i$ is the correct image after receiving $X$ responses. We plot this probability, $P(I_i | #R = X)$, for nine independent HITs published to AMT for the correct candidate image. Figure 6 (a) plots the above probability as a function of the number of responses for each of the nine independent HITs. Each line corresponds to the probability calculated by the system for the correct weed image. Figure 6 (b) plots the average probability (with error bars) calculated by our system for the correct image as a function of the number of responses for the nine HITs. From the figures, we can draw two conclusions. First, for most HITs, the probability of the correct image being selected improves slowly after ten AMT responses. Moreover, fewer than ten responses are insufficient since the transients are too noisy to make reliable inferences. Hence, the system should wait for at least ten responses before making a reliable inference. Secondly, the figure shows that 0.8 is a reasonable threshold to determine when the system should decide that it is confident that the right weed image has been identified. Finally, there are at least two HITs where the probability of selecting the correct weed image is below 0.5. There are several instances, therefore, when the untrained AMT crowd is unable to identify the correct image, and it is important to use extension agents. Since extension agents compare the candidate image with the entire weed database, they can perform accurate weed detection even if the error is in the image processing subsystem. In more than 80% of the HITs, however, it is possible to use the low cost AMT crowd service to get fairly accurate results without any intervention from extension agents.
7.3. System Latency

In our next set of experiments, we evaluate the latency of identifying the correct weed image. We measure the latency of capturing and uploading images to our backend server, analyzing the images using our image processing algorithm, and using Amazon Turk to identify images using an untrained human crowd. Figure 8 (a) presents a stacked bar graph that plots the amount of time taken by different components of the system. We use a threshold of 0.8 to determine when the correct weed image has been identified. The intuition behind the threshold is explained in the previous section. In the figure, we also plot the HITs that did not provide the right answer (HIT 1 and HIT 3) after 40 responses. For the other HITs a threshold of 0.8 was reached in fewer than 35 instance. For each HIT, we perform 20 iterations for the same experiment and report the average. The y-axis is plotted in log-scale. The “Upload time” corresponds to the time taken to upload the images to our backend server and is a function of the backhaul connection bandwidth to the Internet (over Wi-Fi). The “Create Request time” is the amount of time taken to publish a request to Amazon Mechanical Turk, and

This can substantially reduce the cost of detecting weed infestations.
the “AMT Response Time” is the cumulative time of receiving sufficient responses from turkers that the system is confident that a candidate image has been identified. On an average, it takes less than 3 hours to correctly identify the weed image. We have found that the time taken by the image processing algorithm to determine the top five images is 1.9 seconds, which is minuscule compared to the latency associated with the other components of the system, hence the image processing numbers do not appear on the graph. The maximum latency is incurred in receiving responses from the AMT workers. We magnify this incurred time for each HIT in Figure 8 (b) that plots the amount of time taken by the system as a function of the number of responses for each HIT. From the figure, we find that the amount of time taken is proportional to the number of responses and increases nearly linearly for the first fifty responses. It is therefore possible to design a framework that takes a latency constraint as input and determines the number of responses that the system should accumulate before it must make an inference.

7.4. Cost Vs Latency Tradeoffs

An important independent variable in our evaluation is the amount of money spent on each HIT by the system. Specifically, we study the effect of the amount of money spent per HIT on the time to solicit responses from turkers and the amount of time spent by turkers on generating responses. This is an important variable in our system since it helps explore the tradeoffs
between the cost of publishing HITs and the latency associated with inferring the correct answer. We explore two cost rates for publishing HITs—5c per HIT and 1c per HIT. Figure 9 (a) plots a cumulative distribution function of the amount of time taken for a response, and Figure 9 (b) illustrates the cumulative distribution function of the amount of time spent by a turker on a HIT. The lines that correspond to the 5c and 1c HITs are annotated in the graphs. From the graph we find that the median response time for 5c HITs is one-third of the 1c HITs. However, the amount of time spend by turkers on these HITs is equivalent (as shows in Figure 9(b)). To help understand this observation, we present Figures 10 (a) and (b). The figures illustrate the total amount of time taken per HIT, the number of responses, and the cost of publishing the HITs. It is clear that the amount of money spent on the HITs is higher for the 5c case while the total number of responses received are identical—this is because a minimal number of responses are required to make an accurate inference. However, these results also show that turkers prioritize HITs based on the amount of money they earn per HIT. Hence, a HIT with a larger price tag will be attended to by the turker sooner leading to lower overall latency which in turn leads to a higher overall cost. As future work, we plan to utilize this insight to design an optimization framework that can determine the optimal price tag for HITs such that a latency constraint can be met.
7.5. Micro-benchmarks

Finally, we evaluate the overhead of our system with respect to the energy consumed by our smartphone applications and the amount of memory utilized by the phone applications, both for the expert and farmer versions. Specifically, in our experiment we log the battery depletion as a function of time on the Android and iOS platform as we use our crowdsourcing application. Figure 11 illustrates the energy consumed when creating a request at the farmer end and uploading that request. We performed the experiment across twenty different requests. From the figure, we find that the average energy consumed per request is around 20 J which is equal to 1.7 mAh. Hence, a single request on a 1500 mAh battery (common battery size for iPhone and Android phones) can reduce the battery lifetime by 0.1%. Since a farmer is unlikely to upload more than 2-5 requests per day, the energy overhead of our system is low. In terms of memory usage, the Android application uses between 54 MB and 105 MB, and the iOS application consumes between 5 to 20 MB of RAM. Most of the memory overhead is due to the caching that the application performs with images in the database to minimize latency.

Figure 11: Energy consumed in Joules in performing different operations at the smartphone application.
8. Conclusion and Future work

In this paper, we present a hierarchical system that uses smartphone image capture applications, a backend image processing algorithm, and two levels of crowdsourcing to identify weed images. The system provides low latency, low cost, and accurate identification of invasive weed infestations. Such an automated system can help reduce the loses caused by the delay in identifying weeds, and hence, lead to quick remedial control practices applied to contain weed infestations. The two crowdsourcing levels consist of a non-expert inexpensive AMT crowd and a set of expert county extension agents. We propose a probabilistic decision engine that determines, in an unsupervised way, whether inputs from extension agents are required for weed identification or whether the low cost amazon turk crowd is sufficient to identify the weed image. We evaluate our end-to-end system using real weed images and show that the system can provide accurate weed identification within 3 hours while minimally using the extension agents. Moreover, the system can provide highly accurate weed identification. As future work, we plan to use the insights gained from the economics underlying the crowdsourcing approach to design a cost model-driven optimization framework that takes latency and cost constraints and determines the best possible candidate image. Additionally, we plan to leverage additional contextual data associated with the images to improve the machine intelligence (image processing) component of the system.

References


23
URL http://www.mturk.com/


URL http://linkinghub.elsevier.com/retrieve/pii/S0168169910002620

URL http://linkinghub.elsevier.com/retrieve/pii/S0031320307003305

URL http://dx.doi.org/10.1117/12.144030


URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4451360

URL http://asae.frymulti.com/abstract.asp?aid=17244&t=1

URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.6500&rep=rep1&type=pdf


[22] S. Dave, K. Runtz, Image processing methods for identifying species of plants, WESCANEX 95. Communications, Power, ... (95).
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=494064

URL http://linkinghub.elsevier.com/retrieve/pii/S0168169908001981

URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4052483


URL http://portal.acm.org/citation.cfm?doid=985692.985733

URL http://doi.acm.org/10.1145/1814433.1814443
[28] Invasive species definition clarification and guidance white paper.


URL http://doi.acm.org/10.1145/1459359.1459577


URL http://dblp.uni-trier.de/db/conf/oopsla/oopsla2012.html#BarowyCBM12

URL http://0-doi.acm.org.library.uark.edu/10.1145/1869086.1869094

URL http://dx.doi.org/10.1007/BF00992698