A platform for crowdsourcing the creation of representative, accurate landcover maps

Estes, L.D.[a,b,1,*], McRitchie, D.[c,1], Choi, J.[a], Debats, S.[a], Evans, T.[d], Guthe, W.[a], Luo, D.[a], Ragazzo, G.[a], Zempleni, R.[a], Caylor, K.K.[a]

[a] *Civil and Environmental Engineering, Princeton University, Princeton, NJ, 08544 USA*
[b] *Woodrow Wilson School, Princeton University, Princeton, NJ, 08544 USA*
[c] *Computational Science and Engineering Support, Office of Information Technology, Princeton University, Princeton, NJ, 08544 USA*
[d] *Department of Geography, Indiana University, Bloomington, IN 47405 USA*

Highlights

- DIYlandcover crowdsources the generation of landcover data, using human pattern recognition skill to create accurate maps with rich geometric detail.
- It incorporates representative sampling and worker-specific accuracy assessment protocols, and connects to a large online job market. This design addresses three problems with crowdsourced mapping: representativity; data reliability; product delivery speed.
- In a trial case, South African cropland was mapped with 91% accuracy by novice workers. A scaling up analysis found that an Africa-wide cropland map could potentially be developed using this software for $2-3 million within 1.2-3.8 years.

Abstract
Accurate landcover maps are fundamental to understanding socio-economic and environmental patterns and processes, but existing datasets contain substantial errors. Crowdsourcing map creation may substantially improve accuracy, particularly for discrete cover types, but the quality and representa-

*Corresponding author
Email address: lestes@princeton.edu (Estes, L.D.)
[1]Equal contributors

Preprint *November 10, 2015*

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

tiveness of crowdsourced data is hard to verify. We present an open-sourced platform, DIYlandcover, that serves representative samples of high resolution imagery to an online job market, where workers delineate individual landcover features of interest. Worker mapping skill is frequently assessed, providing estimates of overall map accuracy and a basis for performance-based payments. A trial of DIYlandcover showed that novice workers delineated South African cropland with 91% accuracy, exceeding the accuracy of current generation global landcover products, while capturing important geometric data. A scaling-up assessment suggests the possibility of developing an Africa-wide vector-based dataset of croplands for $2-3 million within 1.2-3.8 years. DIYlandcover can be readily adapted to map other discrete cover types.

Keywords: remote sensing, landcover, crowd-sourcing, accuracy assessment, representative sampling, object extraction

Availability

DIYlandcover's source code will be made available free of charge for suitable non-commercial purposes under a GPLv3 license, upon consultation with the authors. For those interested in commercial applications, the prospective licensee should contact Princeton University's Office of Technology Licensing. The details of a specific application of the software for delineating crop fields in sub-Saharan Africa can be found at mappingafrica.princeton.edu, together with associated information about participating in the project, including digitizing rules and links for accessing the mapping interface.

1. Introduction

Regional maps of landcover provide critical information on food security estimates (e.g. Monfreda et al., 2008; Licker et al., 2010; See et al., 2015; Lobell, 2013), models of land-atmosphere interactions (e.g. Liang et al., 1994), and calculations of carbon stocks (e.g. Ruesch and Gibbs, 2008), greenhouse gas emissions (e.g. Searchinger et al., 2015), and habitat change (e.g. Gibbs et al., 2010). These maps are particularly important in developing regions, such as sub-Saharan Africa, where government land use data are often lacking, error-prone, and inconsistent (Ramankutty et al., 2008; See et al., 2015).

These developing regions are also experiencing rapid land use changes (Gibbs et al., 2010; Rulli et al., 2013) that pose pressing development challenges (e.g. how to feed people at substantially lower environmental cost Searchinger et al., 2015).

Unfortunately, landcover datasets derived from medium to coarse resolution satellite sensors are particularly inaccurate (Fritz et al., 2010; Fritz and See, 2008). One major reason for poor accuracy is the fact that land use patterns in these regions are dominated by smallholder farming. Smallholder fields are typically smaller ($\leq$2 ha) than the resolution ($\sim$6 ha) of the most commonly used satellite imagery (Jain et al., 2013). Furthermore, smallholders often plant diverse mixtures of crops, which further increases within-pixel heterogeneity (Jain et al., 2013), and their fields often contain remnant trees and have irregular boundaries, which makes them spectrally harder to distinguish from the surrounding vegetation (See et al., 2015; Lobell, 2013).

New techniques for merging multiple landcover products are helping to substantially improve map accuracy (Fritz et al., 2011, 2015). However, these approaches cannot overcome the mismatch between sensor resolution and smallholder field size. High resolution satellite imagery (<5 m) is becoming increasingly available–and presumably will become more affordable–so the resolution problem should be solved in the near future (See et al., 2015; Lobell, 2013). But high resolution comes at the expense of higher spectral variability; centimeter-scale data require lower orbits, narrower swaths, and greater communication bandwidth, which combine with clouds to greatly limit the area that can be imaged under contemporaneous environmental conditions, and from comparable viewing angles. This means that high resolution image mosaics covering large areas contain substantial and largely uncorrectable spectral differences caused by variations in atmospheric conditions, vegetation phenology, and bidirectional reflectance. This variability propagates error in automated classifications over large regions, which can already be substantial when there is high within-cover variability (Debats et al., 2015), or high heterogeneity among cover types (Gross et al., 2013).

It remains a major challenge to develop algorithms that can accurately classify landcover in the face of both increased image variability and substantial spatial heterogeneity. Promising methods are emerging, however, which draw on advances in computer vision and machine learning, such as semantic segmentation (e.g. Schroff et al., 2008) and Randomized Quasi-Exhaustive feature selection (Tokarczyk et al., 2015), to find optimal classifiers within complex urban environments Frhlich et al. (2013) and highly variable smallholder fields (e.g. Debats et al., 2015). However, these advances are primarily in pixel-wise classification. Accurate, automated methods for extracting individual objects within a single cover type, particularly over wide areas, is arguably even more difficult. Object delineation is an important goal of landcover mapping, as cover geometries encode critical social and environmental information (Fritz et al., 2015), and can play an important role in improving environmental monitoring systems. For example, in agroecosystems, field boundaries can provide a filter for extracting "pure", crop-specific time series of satellite-derived vegetation indices, which helps to improve the accuracy of remotely sensed yield estimates (Estes et al., 2013a,b). Some limited progress has been made with automated approaches, but these have been demonstrated mainly for small areas where the cover objects have regular geometries and sharp boundaries (e.g. commercial agricultural fields Yan and Roy, 2014; Ozdarici-Ok and Akyurek, 2014; Ozdarici-Ok et al., 2015). Such methods are not yet proven over large areas with more complex, less distinct cases.

An alternative approach is to employ humans, who are very adept at recognizing patterns in noisy images (Biederman, 1987). The superiority of human over machine pattern recognition provides the motivation for CAPTCHA (Ahn et al., 2003), which secures websites by requiring human users to recognize fuzzy or irregular letters and numbers that are too difficult for automated algorithms to identify. Human-interpreted landcover maps are thus likely to be consistently more accurate than automated classifiers. Unfortunately, since humans are much slower at data processing than computers, human-generated landcover maps covering large areas will require much more time and expense to create. However, this problem is being alleviated by the growth of the internet, which makes it increasingly feasible to turn pattern recognition problems into many small tasks that are undertaken by a large number of online workers—the human equivalent of parallel processing. This ability to "crowdsource" (Howe, 2006) such work supports projects ranging from galactic classification (Lintott et al., 2008) to ornithological surveys (Sullivan et al., 2009). Crowdsourcing of landcover is already being used in the Geo-wiki project, which uses online volunteers to correct landcover data based on their own interpretations of high resolution satellite imagery (Fritz et al., 2009, 2012, 2015). Recently, these data have been used to create the most accurate (82%) global cropland map (Fritz et al., 2011, 2015).

While the use of crowdsourcing is an extremely promising development for landcover mapping, and is being increasingly used for this and other en-

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

vironmental monitoring applications (Jacobson et al., 2015; Fraternali et al., 2012; Schellekens et al., 2014), many existing projects (e.g. OpenStreetMap (openstreetmap.org)) are geared towards users who create content according to their personal interests, thus the resulting maps are unlikely to be geographically representative (Fraternali et al., 2012). Furthermore, verifying the accuracy of crowdsourced data is a challenge (Allahbakhsh and Benatallah, 2013; Flanagin and Metzger, 2008; See et al., 2015) that remains largely unaddressed by existing platforms. In terms of using crowdsourcing to improve landcover data, prior efforts have focused primarily on validating pixel-based classifications, and less on delineating individual cover objects, which is arguably one of the greatest advantages that people have over machines. Indeed, recognizing and digitizing individual, discrete cover types such as crop fields is considered fairly "straightforward" for humans (Yan and Roy, 2014).

In this paper, we describe ***DIYlandcover*** (or "Do-it-Yourself" landcover), a new platform for creating crowdsourced landcover data that addresses the three aforementioned limitations. DIYlandcover was designed for mapping discrete, but "noisy", cover types, where object extraction is of primary interest. Specifically, our platform provides online workers with tools to 1) delineate landcover objects within 2) representatively selected locations, while the resulting maps are subjected to 3) periodic quality assessments that provide estimates of individual worker and overall map accuracy. We provide an overview of DIYlandcover's design and mechanics, and report on the results of a trial application mapping crop fields in South Africa, which suggests that DIYlandcover allows inexperienced online workers to generate high accuracy (>90%), geometrically rich, and geographically representative landcover data at a much faster rate than is usually possible with human-based mapping.

2. System design

The inspiration for DIYlandcover came from GeoTerraImage, a company that mapped South Africa's arable cropland by manually digitizing fields visible in high resolution satellite imagery (GeoTerraImage, 2008). The resulting map set is 97% accurate in distinguishing cropped from uncropped areas at a 4 ha resolution (see detailed accuracy assessment in Appendix S1), and provides rich detail on field type and geometry. However, making these maps was an expensive and lengthy process; the estimated labor cost

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

for digitizing was \$5 km^{-2}, and the project took approximately 2.5 years to complete (Ferreira, pers. comm.).

We developed DIYlandcover to help overcome these constraints of cost and production time, while retaining the advantages of human image interpretation skill demonstrated by GeoTerraImage. Our platform connects workers in an online job marketplace to a map application programming interface (API) that hosts high resolution satellite imagery. DIYlandcover currently works with Amazon's Mechanical Turk (Services, 2012) and the Google Maps API, but these could in principle be replaced by other services. These two aspects of DIYlandcover substantially reduce both mapping costs and completion times, because the imagery is free and the platform can access a potentially large number of workers.

Given the distributed and anonymous nature of the online job market, we cannot intensively train workers (as GeoTerraImage did), yet our mapping task is complex, requires significant image interpretation skill, and must be completed in a systematic manner. Therefore, to ensure the scientific quality of its maps, DIYlandcover incorporates site selection and accuracy assessment protocols (Fig. 1). A sampling grid (SG in Fig. 1) over the desired study region provides the basis for collecting stratified random samples. The first draw identifies sites where the researcher/administrator (the "Requester"; Allahbakhsh and Benatallah, 2013) will provide landcover reference maps (black cells). Subsequent draws select sites where workers will create new maps (grey cells). This sample of locations is then sent to the job marketplace. All workers must pass an initial qualification test (Q1 in Fig. 1) that proves their ability to map a handful of sites with a minimum level of skill. Once qualified, workers begin mapping. Each worker will map both grey and black sites, which are respectively referred to as N (for normal) and Q (for quality assessment) sites. Q sites are indistinguishable from N sites, and are intermingled such that each worker has a Requester-defined probability of encountering a Q site. Completed maps from N sites are inserted into DIYlandcover's database (D), while maps from Q sites are first scored according to their agreement with their reference maps (Q2 in Fig. 1). Maps that fall below a minimum score are rejected. Map scores are incorporated into a worker-specific quality score, which is used to assign confidence to all maps generated by a worker, and to determine overall map accuracy. Workers are paid (P in Fig. 1) for each site mapped, with the possibility of bonus payments linked to quality scores.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

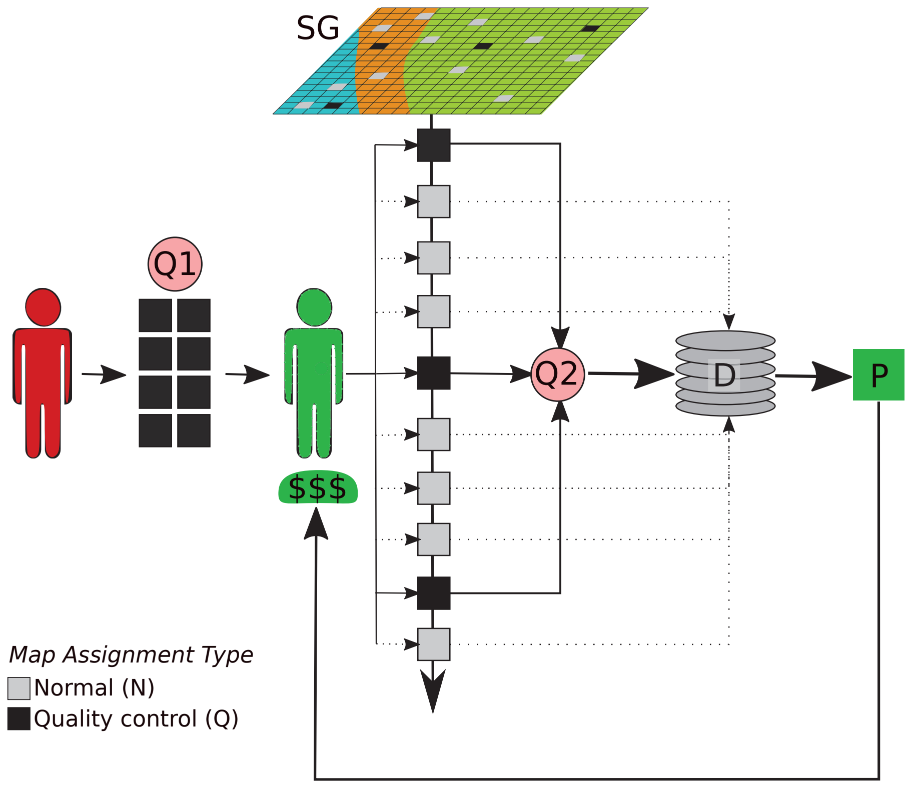

Figure 1: An overview of DIYlandcover's design. A survey grid (SG) is overlaid on a geographic area, and then random samples (weighted proportionally to the probability of cover type presence, represented by green, orange, and blue) are drawn to specify where groundtruth maps will be generated (black cells) to support worker map quality (Q) assessment. Subsequent random draws (grey cells) select sites that are undertaken as normal (N) mapping assignments. N and Q sites are sent inter-mingled to the online job marketplace for mapping. A first time worker (red) must take an initial map qualification test (Q1), after which she or he is qualified (green) and begins mapping. Maps from N sites are stored in the database (D); Q site maps are first scored based on their agreement with groundtruth (Q2). This score contributes to a longer term worker quality score, which is used to assess overall map quality and allows performance-based bonuses to be paid on of fixed per site payments (P).

3. The mechanics of DIYlandcover

The basic structure of DIYlandcover consists of three elements (Fig. 2): the main server hosting DIYlandcover's database, here a Linux virtual ma-

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

chine with PostgresSQL (9.4) with the PostGIS (2.1) spatial extension; a map server hosting the satellite imagery, in this case the Google Maps API (Developers, 2012); the online job market, Mechanical Turk (Services, 2012). Within this structure several key processes govern the creation and management of mapping tasks.

3.1. Site selection

A "master grid" covering the study area is first created as a PostGIS table. Each cell provides a unique identifier, and the cell resolution defines the area of an individual mapping task. This grid is intersected with a second grid containing landcover occurrence probabilities, which are converted into categorical weights. A third field is created that indicates whether each cell is available to be mapped or not.

After the initial random draw (of a user specified size) is taken to identify quality assessment (Q) sites (Section 2, Fig. 1), the selected cells' status is set to unavailable. The geometries are written to individual keyhole markup language (KML) files, and their IDs are added to a "KML data" table, where a field specifying cell type is set to "Q" to indicate that the corresponding KMLs reference quality control sites. The user has to provide landcover reference maps for these sites, the geometries of which are stored in a "reference maps" table.

The next draw collects sites that will form the normal ("N") map production process, where a worker (or workers) creates maps for locations where the underlying landcover is unknown. This step is governed by *KMLGenerate*, an R process that connects to the database (via the RPostgreSQL package; Conway et al., 2012), takes a weighted random draw of size X (a parameter stored in the "configuration" table that holds all variables used by DIYlandcover) from the master grid table, writes each cell geometry to a separate KML file, adds the selected cell IDs to the KML data table, and sets the field type value to "N". The script changes the cell status in the master grid to unavailable. As N type maps assignments are completed, their status is set to mapped in the KML data table. *KMLGenerate* runs as a daemon, selecting a new random draw as soon as the number of unmapped sites falls below a specified number, ensuring that there is never a system delay in sending mapping assignments to the job market (see 3.2).

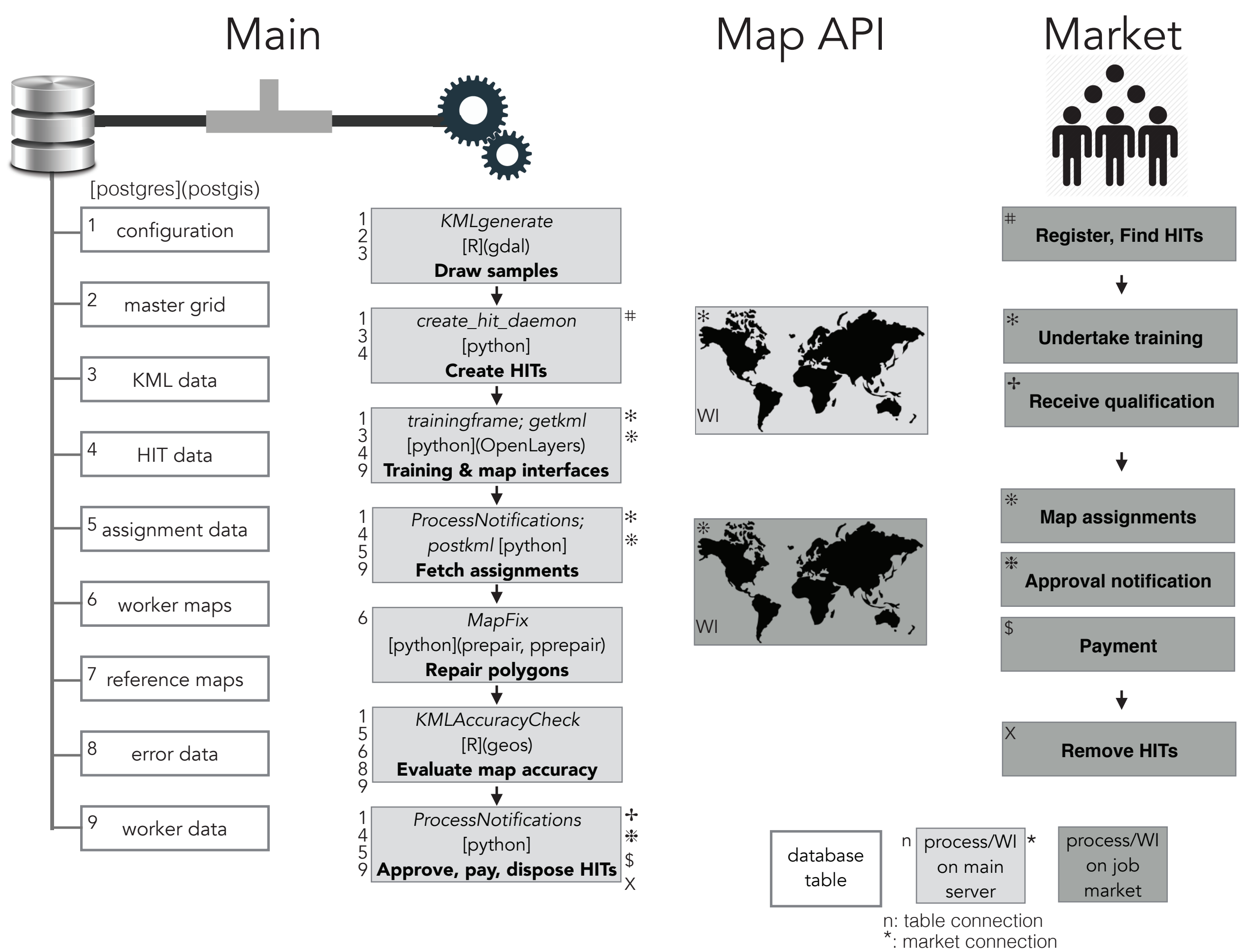

Figure 2: The components, and primary processes of DIYlandcover. The main server contains the system database and processes. Primary data tables are shown by the white boxes with grey borders. Primary processes are shown in light grey boxes (process names are italicized, primary software in brackets and its external dependencies in parenthesis, and description in bold). Server processes interact with specific data tables (indicated by the numbers to the left), and with processes that occur in the online job market (indicated by symbols to the right). The two versions (one for training, one for qualified workers) of the worker interface (WI) to the map API are shown, color-coded according to where they are hosted (on main server or online market).

3.2. Creating mapping assignments

Following selection, each site is converted into a mapping task for online workers. These tasks are referred to as Human Intelligence Tasks (HITs), in Mechanical Turk's parlance. HITs are created by (*create_hit_daemon*), a python daemon that uses the boto library to interface with Mechanical Turk (MT). The daemon polls MT (at regular intervals) to see how many DIY-landcover HITs of types Q and N exist on MT (zero at start of production), and whether they fall below their minimum required numbers. These numbers are calculated from two configuration parameters: the minimum total number of HITs that should be available on MT, and the percentage of these that should be of Q type. If the actual numbers of each type fall below their target numbers, *create_hit_daemon* selects the IDs of available KMLs from the KML data table, and sends these together with associated HIT metadata, which includes the pay rate, the number of times the HIT should be mapped, the qualifications required to undertake the HIT (see 3.5), and a definition of the task. MT then registers each HIT and provides it with a unique HIT ID and registration time, which is logged into a "HIT data" table on the main server.

3.3. Undertaking the mapping assignment

Once a HIT is registered on MT, it is visible to all workers in the marketplace, but can only be undertaken by qualified workers (see 3.5). Qualified workers who choose to undertake DIYlandcover-generated HITs are first shown a default HIT preview, and they must choose to accept it before they can see the actual location to map. This step helps prevent workers from declining more challenging sites, which bias the sample towards simpler land-covers.

To enable workers to perform a mapping HIT, DIYlandcover uses an OpenLayers interface to the image server, which sits within MT's user screen, centers the map view on the site of the HIT location, and provides a set of digitization tools (Fig. 3). As soon as the worker accepts the HIT, it becomes a mapping assignment that is issued a unique assignment ID. A Web Server Gateway Interface (wsgi) script, *getKML*, retrieves the OpenLayers javascript, the frame size parameters for the MT interface, the url for the KML demarcating the sample site, and user instructions (e.g. tool use tips), and passes these to MT, and collects the worker, assignment, and HIT IDs and acceptance time, and records these into the "assignment data" table.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

The worker then draws polygons around the landcover type(s) of interest that intersect the KML sample square (Fig. 3), and has the option to edit or delete individual geometries and provide comments. On completion, the worker saves the map, and is then taken to the next HIT preview screen. Alternatively, the worker may choose to return the assignment uncompleted. If this happens more than a specified number of times, the worker's qualification can be revoked (see 3.4), which is another check against sample selection bias. The assignment is automatically abandoned if it is not completed within a defined time. We impose this last restriction to minimize bias in the estimation of wage rates (see 4.2); if workers leave the assignment unfinished on their computer for long periods, the amount of time required to complete assignments will be inflated.

When the assignment is completed, returned, or abandoned, MT sends an email notification to the main server, where it is retrieved by *ProcessNotifications*, a python process. If the assignment is returned or abandoned, it is marked as unprocessed and returned to the pool of available HITs on MT, and the worker receives no pay. If the assignment was completed, post-processing routines are triggered.

3.4. Processing completed assignments

Several processing steps must be performed before the worker is paid for the completed assignment, which depend on whether the worker created any polygons during the assignment, and whether it was of Q or N type. If the worker created polygons, then the geometries, KML ID, assignment ID, and completion time are stored in the "user maps" data table by process *postKML*, which then triggers *mapFix*, a python script that invokes prepair and pprepair (Ohori et al., 2012), which repair the topologies of single and multi-polygons, respectively. This step is essential because hand-digitized polygon data often contain errors, such as self-intersections and unintended overlaps, which can render topologies invalid and cause subsequent spatial analyses (per 3.4) to fail. The repaired geometries are then inserted into the user maps table.

Next, the assignment is given a score, which is recorded in the assignment data table. If the assignment was of N type, this score is null; for Q type, *KMLAccuracyCheck*, an R process, is called to compare the worker's and reference maps, with the score determined by:

$$\mathrm{S} = \beta_1 \mathrm{C} + \beta_2 \mathrm{O} + \beta_3 \mathrm{I} \tag{1}$$

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

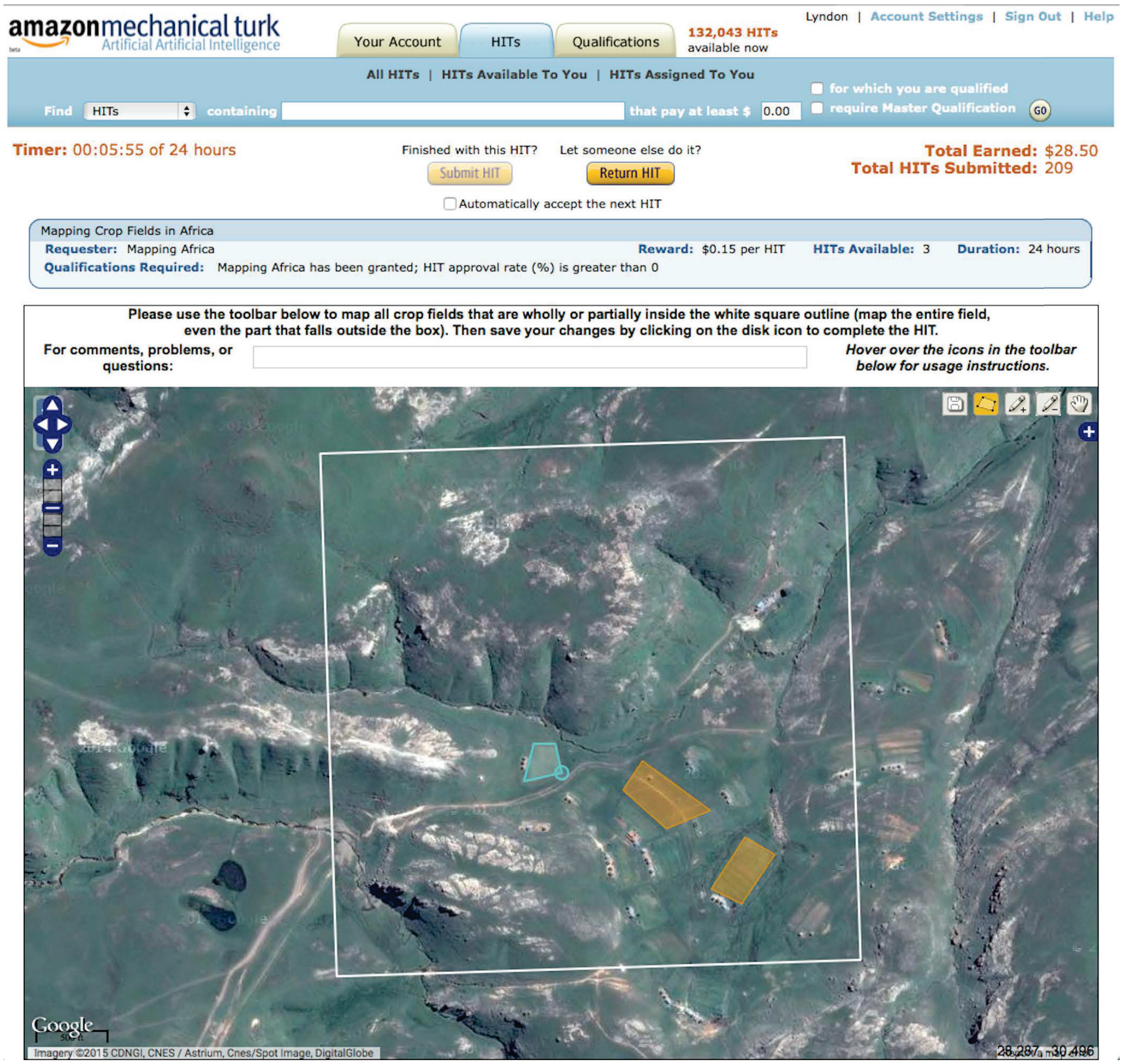

Figure 3: The DIYlandcover mapping interface within Amazon.com's Mechanical Turk job marketplace. The white square is the KML sampling frame, gold polygons are completed crop field polygons, the blue polygon is a field in the process of being mapped. Mapping controls are in upper right corner of the image frame.

Where S is overall mapping accuracy, β_1-β_3 are user-defined weights, and:

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

$$C = 1 - \frac{\mathrm{abs}(n - N)}{\max(n, N)} \tag{2}$$

$$O = \frac{a}{a + c} \tag{3}$$

$$I = \frac{A + D}{A + B + C + D} \tag{4}$$

Or:

$$I = \left(\frac{A}{A + C} + \frac{D}{B + D}\right) 0.5 \tag{5}$$

With C being count error, or the agreement between the number of landcover polygons in the worker's maps (n) and in the reference data (N). O measures map agreement for those parts of the worker's and reference polygons that fall *outside* of the KML grid, where a is the area of overlap, and c is the false negative error (i.e. the area of reference field polygons falling outside the grid that the worker failed to map). I measures map accuracy *inside* the KML grid, with A being the grid interior equivalent of a, B the false positive error (i.e. landcover incorrectly labelled by the worker), C the false negative error (landcover area missed by the worker), and D the true negative area (area correctly left unmapped). I can be calculated using standard classification accuracy (Eq. 4), or a variant of the True Skill Statistic (Eq. 5 Allouche et al., 2006), a more stringent measure that corrects for class prevalence, which we compressed to fall between 0 and 1 rather than -1 to 1. The areas of a, c, A, B, C, D are calculated using intersection and difference operations provided by the rgeos library (Bivand and Rundel, 2013), after transforming maps to a projected coordinate system.

We include the O metric to encourage workers to completely map features intersecting the sampling grid (i.e. either falling entirely within or both within and outside of it), in order to have unbiased estimates of landcover size classes. However, we can only partially assess the accuracy of exterior features because it is impossible to correctly define negative space outside the sample grid, since it is both unbounded and may contain target features that will not be mapped because they do not intersect the grid. An error map showing each of the accuracy assessment components is illustrated in Figure 4.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

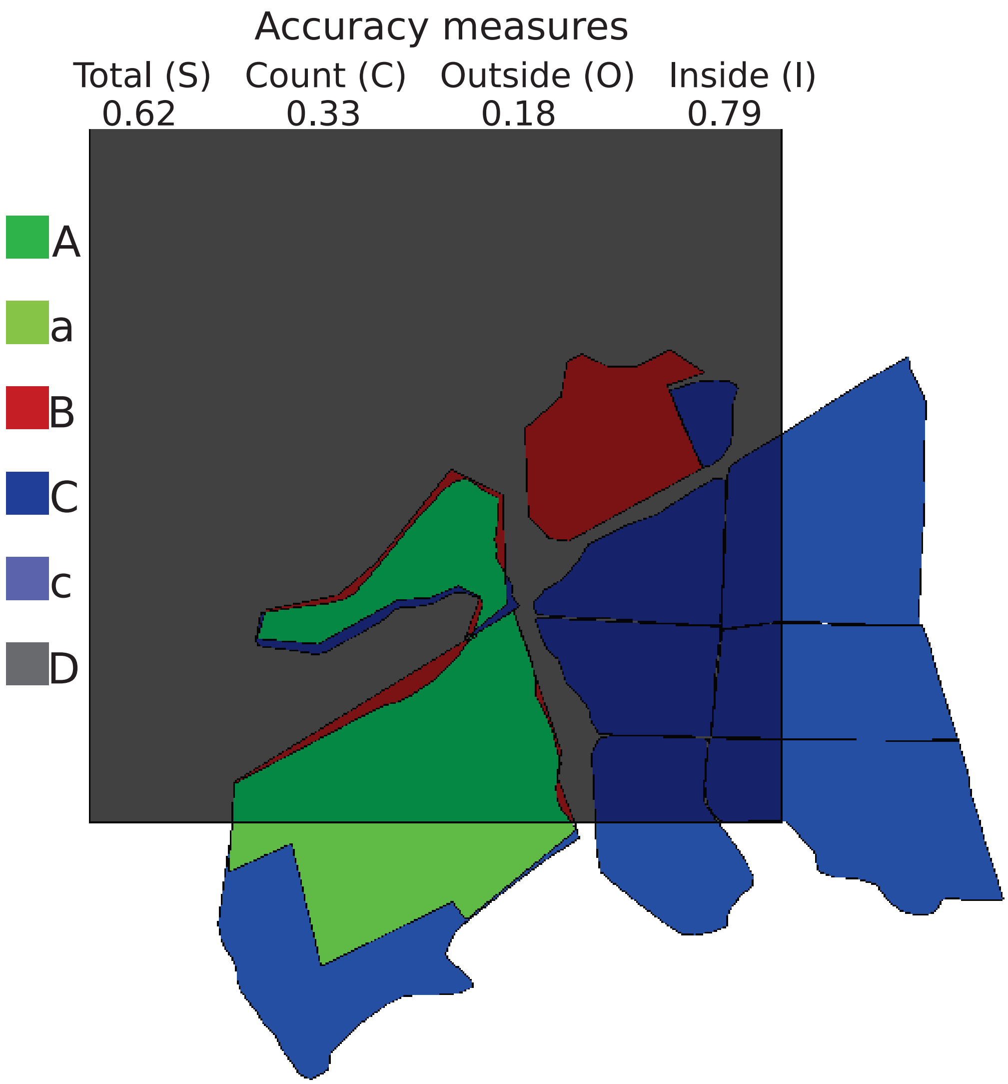

Figure 4: A graphical illustration of the accuracy assessment algorithm (as applied to cropland maps), providing the resulting scores for overall accuracy (Eq. 1) and count, outside, and inside error (Eqs. 2-5), where each component ranges between 0 (most error) and 1 (no error). The area of each error component is color-coded: A (agreement inside the grid), a (agreement outside), B (false positive error inside the grid), C (false negative inside), c (false negative outside), and D (true negative inside).

Once the algorithm has run, all accuracy measures (S, C, O, I) are stored in the "error data" table, while S is stored in the assignment data table. S is also added to a vector of Q scores for the specific worker (stored in the "worker data" table), which is used to calculate a moving average of the worker's recent performance. If S is above a minimum accuracy threshold, then the assignment is approved. If rejected, then payment is withheld, and a notice is sent to MT where it is added to the worker's system-wide rejection rate. Successive rejections can result in the revocation of mapping qualifications if a worker's *quality* score drops below the accuracy threshold. The quality score is:

$$\text{quality} = \frac{\mathrm{S_i} + \mathrm{S_{i\text{-}1}} + \ldots + \mathrm{S_{i\text{-}(j\text{-}1)}}}{j} - \beta_4 \frac{\mathrm{R_i} + \mathrm{R_{i\text{-}1}} + \ldots + \mathrm{R_{i\text{-}(j\text{-}1)}}}{j} \tag{6}$$

Where i is the most recent S value calculated, and j the total number of recent S scores to use in calculating a mean S. To minimize assignment selection bias (see 3.3), an additional penalty, the worker's rate of assignments accepted but returned without completing (R, which equals 1 for a return, 0 for a completion), is multiplied by a weight β_4 and deducted.

In cases where the worker returns no maps for a Q type assignment, map storage and cleaning does not occur before *KMLAccuracyCheck* is run. In these cases, the C and O scores (Eq. 2 & 3) reduce to 1 where the reference map has no landcover polygons, or 0 if it does. If the assignment is of N type, it is scored as NULL and added to the assignment data table.

Unlike the Q type, N assignments are automatically approved, under the logic that the worker's quality score at the time of map creation is indicative of that map's accuracy. The exception to this is N assignments created by a newly qualified worker (see 3.5), which are marked as "untrusted" in the assignment data table until that worker completes as many Q assignments as are needed to calculate the moving average accuracy score. Upon assignment approval, *ProcessNotifications* relays a message to MT and the worker is paid (see 3.6) from the Requester's account, and then removes the corresponding HIT from MT. Q sites will be re-created as HITs multiple times, while N sites are mapped just one time.

3.5. Worker qualification and payments

All workers performing mapping assignments must first be qualified, which is treated as a special case of Q type assignments. MT evaluates the qualification status of each worker attempting to access a DIYlandcover HIT. If

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

the worker is not qualified, a link to a training module is presented on the MT interface. The module, which is hosted on the main server, is managed by *trainingframe*, a python process, which issues each new trainee a unique training ID. The trainee first watches video tutorials explaining the project and its mapping rules, and is then required to map several training sites, the accuracy of which is assessed by *KMLAccuracyCheck*. Trainees must map each site to the minimum accuracy standard, but are given unlimited chances to do so. A separate set of tables mirroring those used for collecting map, assignment, worker, and error data is used to record training data. Once a worker successfully completes all training sites, a qualification request is posted on MT. A daemon, *process_qualifications_requests*, polls MT at specified intervals, collects these requests together with associated worker and training IDs, examines for each worker whether all training sites were completed successfully, and, if so, adds the trainee's worker ID to the worker data table, sets the qualification status to true, then sends a notice to MT that the worker is now qualified. Candidate workers who fail to pass all training sites, or workers whose qualifications are revoked due to poor accuracy (see 3.4), can repeat the training to qualify/re-qualify.

Upon qualification, workers are paid a small bonus, and can begin mapping assignments. Workers are paid a flat rate for approved assignments. To incentivize worker performance, DIYlandcover also allows bonus payments to be made based on the worker's accuracy score. If implemented, the bonus algorithm, managed by *ProcessNotifications*, pays an extra per assignment amount if the worker's quality score exceeds certain thresholds.

4. Applying DIYlandcover to map South African crop fields

We examined the capabilities of DIYlandcover by applying it to map crop field boundaries in South Africa. South Africa was a convenient test case because it cropland was already mapped (see section 2; GeoTerraImage, 2008) using similar methods, providing both an objective means for evaluating DIYlandcover's performance, and a readily adaptable source of reference maps. Furthermore, South Africa's diversity of agricultural systems are representative of the image interpretation challenges facing workers. This mix ranges from hard to detect communal and smallholder agriculture, to more easily discerned industrial fields (Hardy et al., 2011). South Africa also provides the test site for the Mapping Africa project, which aims to create high quality cropland maps for sub-Saharan Africa.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

4.1. Mapping set-up

We created a 1X1 km, Albers Equal Area Conic-projected sampling grid for South Africa, and used logistic regression to model the probability of cropland presence throughout the country. Equally sized random draws of points selected inside and outside GTI field boundary polygons provided the positive and negative responses of the dependent variable, while predictors were derived from gridded rainfall and elevation data and a map of protected areas (for further details on these variables see Estes et al., 2013b, 2014). The resulting probability was divided into quartiles, which provided the weights used by *KMLGenerate*.

For Q sites, we used the modeled cropland probability categories to draw a random select 609 grid cells (0.05% of South Africa's area), providing a representative sample of South Africa weighted towards agricultural areas. We intersected these with the GTI polygons to create the associated Q data tables (3.1). These polygons were then further edited to make the Q maps consistent with imagery in the Google Maps API, and to conform with the specific mapping rules that we set for workers (Table 1). Workers were asked to map sites where crop fields were actively or very recently (i.e. within the past 2-3 years) used for arable agriculture. This category of agriculture takes many forms in South Africa (see Appendix S1 for an illustration), ranging from large, clearly defined, commercial fields to less geometrically distinct smallholder fields, which often contain trees and mixed crops. Long term fallows, tree crops (orchards, commercial afforestation), and non-agricultural areas were left unmapped. In cases of uncertainty (e.g. the worker had trouble telling whether the field was active or abandoned), workers were asked to map every second field. On top of the high variability in arable fields, narrowing the mappers' focus actually made the task more challenging, because the agricultural types described in Table 1 often look similar, which increases the risk of both false positive and false negative errors. For instance, it is often difficult to tell whether a field is active or abandoned, while young orchards or recently cleared forest compartments can be mistaken for arable fields. In all these examples, field boundaries tend to be clearly visible, thus more inclusive mapping rules would likely reduce both types of error.

The system was set to make random draws of 500 N sites from the master grid each time the number of N sites available for mapping fell below 500 (3.1), in order to ensure that no system latency occurred as the system selected new mapping locations. At least 10 HITs, 80% N and 20% Q, were maintained on MT at any given time, with the system polling MT every 10

Table 1: Rules for mapping crop fields in the South Africa-focused application of DIY-landcover. Workers were asked to map only currently active (i.e. farmed within the past 2-3 years) annual crop fields, and to not delineate other agricultural types.

Feature type	**Action**
No cropland visible	Don't map
Active annual crop field	Map
Fallow crop field	Don't map
Unsure if active crop fields	Map every second feature
Permanent tree crops (orchards/plantations)	Don't map
Improved pastures	Don't map

seconds to see if new HITs were needed (3.2). This relatively low number allowed rapid cycling of HITs through MT, while the ratio of N:Q HITs ensured that worker accuracy was assessed (3.4) frequently during the trial (once in every five assignments).

The accuracy algorithms (Eq. 1) β terms were set as 0.1, 0.2, and 0.7 for the C (Eq. 2), O (Eq. 3), and I terms (here Eq. 4). We selected a low weight for C because determining the boundaries of individual fields from overhead imagery is fairly subjective, even for expert observers, and we did not want to unduly penalize workers for a difference in judgement, yet we also wanted to discourage rapid mapping that erased boundaries between clearly distinct fields. We gave O a slightly larger weight to stress the importance of completed fields that extended outside the sample grid, but a larger weight would give the worker too much credit for cases where no fields intersected the grid. The I term was weighted most heavily because it is the only place where workers' abilities to correctly distinguish null space can be assessed. We used the same weights to assess assignments within the 8-site training module.

Payment was set at \$0.15 per assignment. A four-tier bonus payment algorithm was also written into logic. We did not implement this logic in our initial trial, in order to first assess whether the base rate would allow workers to achieve our target wage of \$8-10 hour^{-1}, but we evaluated the cost implications of bonus payments set to \$0.01, \$0.02, \$0.03, and \$0.05 for worker quality scores exceeding 0.85, 0.95, 0.975, or 0.99, respectively.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

4.2. Trial results

The Mapping Africa trial ran on Mechanical Turk for 26.4 hours between October 2-3, 2013, resulting in 945 mapping assignments, of which 882 were approved, 10 were rejected (due to failing accuracy scores), and 53 were not completed (i.e. returned or abandoned). A total of 707 N sites with 216 (31%) containing worker-delineated polygons were mapped, as well as 185 Q sites, with 65 (35%) having fields (Fig. 5).

These sites were mapped by 15 different workers, from a pool of 18 who passed the initial qualification test. A further 18 took the qualification test but failed to pass. The distribution of mapping effort was highly skewed, with three workers completing 65% of the total assignments (Fig. 6A). The average Q:N assignment ratio for each worker was 18%, but there was high variability among workers who completed less than 50 assignments (Fig. 6A). The mean accuracy assessed across all Q sites (using Eq. 1 with Eq. 4) was 0.91 (out of 1), but Q sites containing fields were mapped with lower overall accuracy (0.79) than sites without fields (0.97; Fig. 6B). Using just the inside component of the score (Equation 4), accuracy was higher for sites with fields (0.89 with fields versus 0.99 without). To understand these accuracy discrepancies more fully, the number of polygon vertices in the reference polygons can be used as proxy for cropland complexity, and thus assignment difficulty. Worker accuracy declined significantly, albeit weakly ($p<0.048$), in relation to this complexity (Fig. 6C). Worker effort also declined strongly as a function of map complexity (Fig. 6D); the more fields there were to map—or the more intricate their boundaries—the fewer vertices placed by workers, presumably to minimize mapping time. This reduction in effort may partially explain the increased error.

Replacing Equation 4 with Equation 5 (the True Skill Statistic; TSS), which corrects for class prevalance (Allouche et al., 2006), to calculate map score (Eq. 1) removed the significant negative relationship between map score and complexity (F-statistic: 0.98; $p<0.32$). At sites with only a few fields, which are both less complex and typically having a much higher share of non-cropped than cropped area, Eq. 4 was more lenient than at more complex sites, because the worker received proportionally more credit for "mapping" the uncropped space. This tendency is seen in Fig. 6E, which shows that map scores calculated using Equation 4 were generally higher than those assessed using Equation 5; for sites with fields, scores were on average 0.1 higher, and up to 0.14 greater where fields were relatively simple

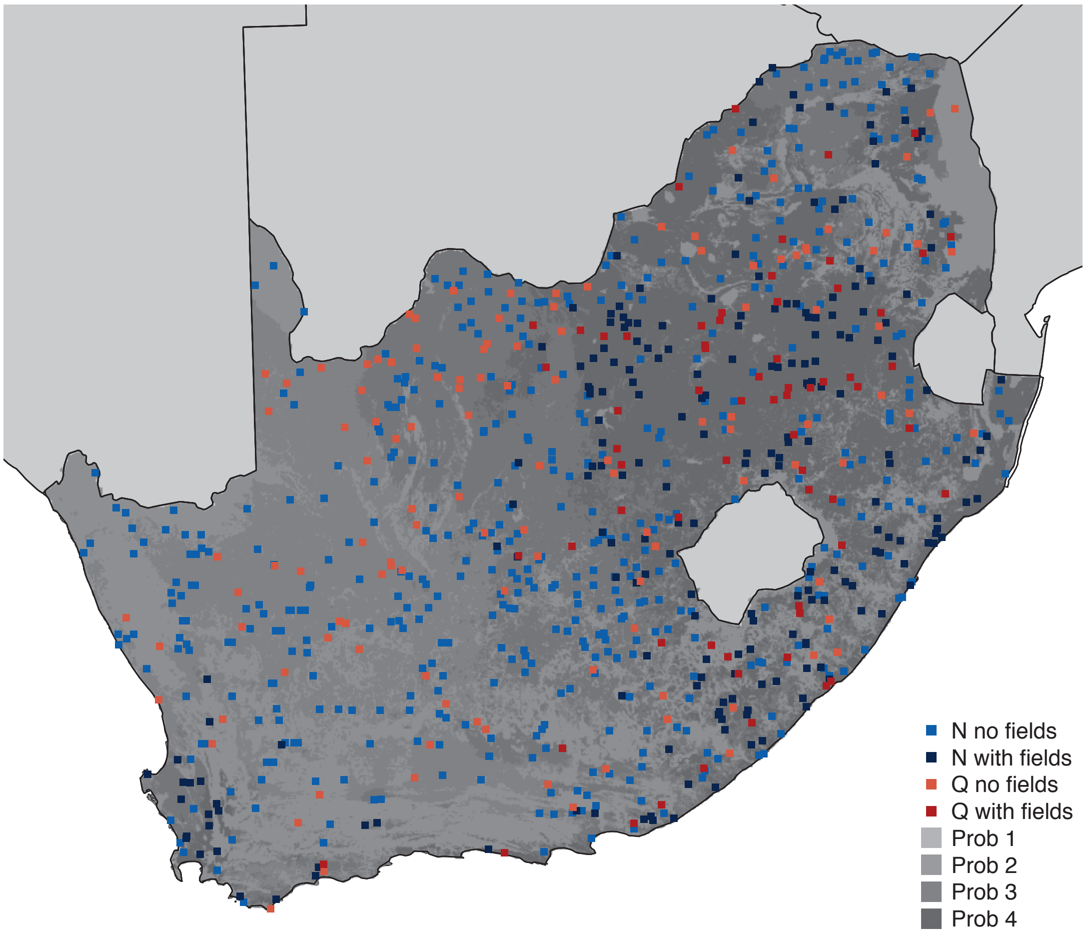

Figure 5: A map of DIYlandcover trial results over South Africa, showing the distribution of mapped sites, color-coded according to their assignment type (Q or N) and whether they contained worker-mapped crop field boundary polygons. The grey shading indicates the four-category weighting derived from a logistic regression model of cropland occurrence.

(i.e. where truth maps had <25-50 vertices, indicating both low complexity and small areas).

Accuracy appears to improve with experience, as workers' average accuracy scores increased in proportion to the number of Q assignments completed. Accuracy gains increased rapidly below 20-25 completed Q assignments, after which they leveled out between 0.9 and 1 (Fig. 6F).

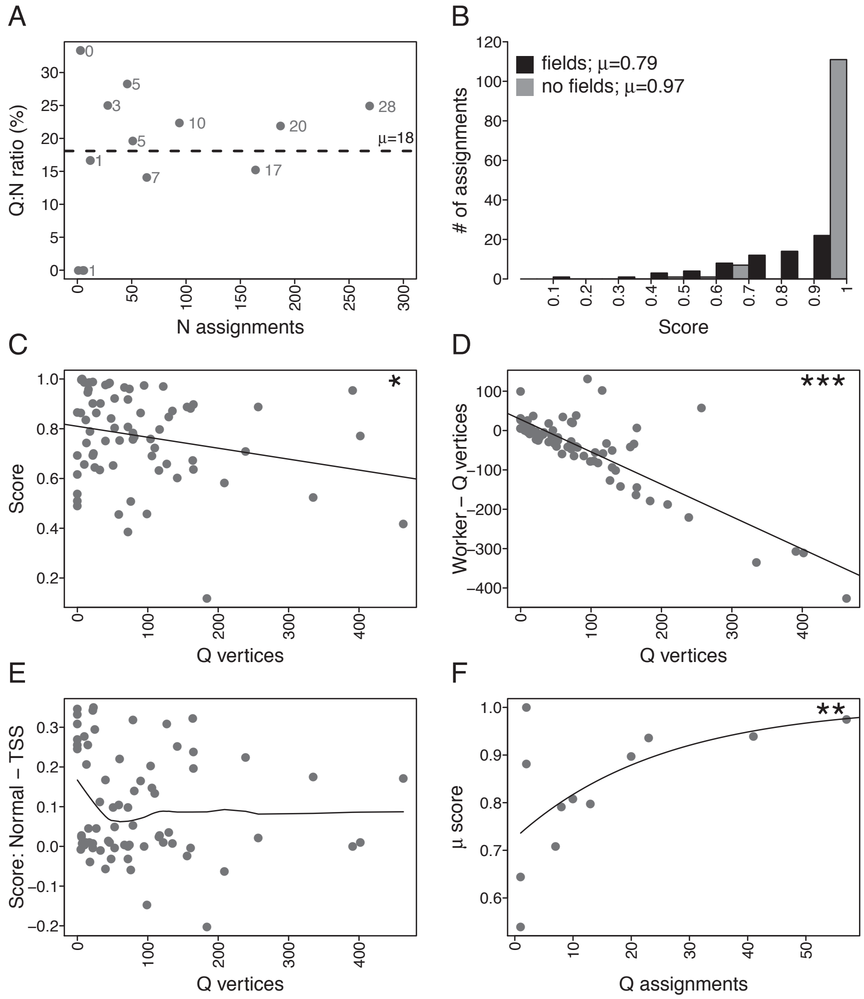

Figure 6: Results from initial DIYlandcover trial, including A) the total number of completed assignments per worker versus the ratio of Q to N type assignments (values in grey next to points represent the percent of total assignments); B) the distribution of accuracy scores segregated by assignment type (black bars = Q type, grey bars = N type); the number of vertices in reference map polygons versus C) accuracy score, D) the difference between the number of vertices in workers' and reference map polygons, and E) the difference between accuracy scores calculated using Equation 4 or Equation 5 (the True Skill Statistic); F) the number of Q assignments completed by each worker versus worker mean accuracy score. Significance of regression fits in C, D, F are: $*p<0.05$; $**p<0.01$; $***p<0.001$. C and D are linear models, F is asymptotic regression.

We paid \$132.30 to workers for the 882 approved assignments, with a total cost to the project of \$145.53 after accounting for Amazon.com's 10% Requester surcharge. Of this, \$28.88 was paid for the 175 approved Q assignments and \$116.66 for the 707 N assignments. Our post-hoc application of the bonus algorithm, which requires workers to complete at least five Q assignments (8 of 15 met this requirement), would have added \$21.89 (15%) to the trial's cost.

To examine the effective worker wage (i.e. the amount the worker would expect at these rates assuming constant, uninterrupted work), we divided total pay by the mean assignment duration, calculated as the difference between assignment acceptance and completion times. Since workers could accept assignments without immediately completing them (maximum assignment duration was 24 hours), we could not precisely measure mapping time. However, our experience suggests that the most complicated sites require <30 minutes of mapping effort, thus we excluded any assignments taking longer than this. The resulting average effective wage was \$10.80 hr^{-1} across all sites, but just \$3.26 hr^{-1} for sites having fields compared to \$13.40 hr^{-1} for sites without fields. Factoring in bonus payments, these would have been \$11.65 overall and \$3.55 and \$14.53 for sites with and without fields (Table 2).

The flat rate cost to map a single square kilometer was \$0.165, including the cost of accuracy assessment and Amazon.com's fees, or \$0.19 had we included bonus payments.

4.3. Estimating the costs of scaling up

We used the time and cost results from the trial to estimate the potential costs of mapping larger regions, in terms of worker payments and total mapping time, using two different payment models. One models used fixed base rates (as in our trial), the other variable rates linked to potential mapping effort, and in each case we tested two different levels of payments: for the fixed case we used rates of \$0.15 and \$0.05[2]; for the variable rate, payments rates were set using the following formula:

[2]Estimated as the approximate difference between US and South African minimum wages, http://businesstech.co.za/news/international/87614/minimum-wage-in-south-africa-vs-the-world/

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

$$R = \$0.01 + (\sum_{w=1}^{n} -1)I \qquad (7)$$

Where R is the rate, w is a categorical weight derived from a map of cropland probability, and I is an increment, set here to \$0.07 and \$0.023 for the higher and lower pay models, respectively (see Appendix S3 for methods). For the cropland weights, we converted the GeoWiki 1 km^2 cropland percentage map (Fritz et al., 2015) into a 10 category map (where 1 = 0-10% cropland and 10 = 90-100% cropland). This map provided a more finely resolved set of weights than our four category map, and covered the entire continent. Since cropland percentages correlate positively with field area and number (albeit with area), these weights also provided a proxy measure for likely mapping effort. We confirmed this assumption by extracting the new weights corresponding to the areas mapped, and used them in a least squares regression to model workers' observed mapping times (R^2=0.1, $p<0.0001$; Appendix S3). The map weights were extracted into a reordered vector using DIYlandcover's weighted random sampling protocol (see 3.1), and then used with Equation 7 to assign payments for each site. We added the trial mean bonus rate (\$0.023) onto these payments (and to the fixed pay rates), then calculated the cumulative cost for mapping all 29,924,000 km^2 of Africa for each pay model, multiplied by 1.4 to represent 1) an additional 10% of mapping effort for quality assessment, and 2) administrative costs of 30%.

To estimate the total time required to map the continent, we used the predicted mapping times resulting from the regression model. The model was run 1000 times on random subsets of the data to obtain prediction uncertainties for each weight level, from which the mean, 2.5th, and 97.5th percentile values were extracted. These predicted time values were assigned to their corresponding weights in the reordered weights vector, from which mean, upper, and lower estimates of cumulative mapping hours were calculated (Appendix S3). We then created three hypothetical models of worker involvement, in which either 100, 250, or 500 workers, each mapping one hour per day, undertook the work, and used the resulting daily total mapping hours to convert the cumulative mapping time into estimates of how long it would take to map the entire continent (in years).

The cost model results show that variable pay rates would be considerably more efficient than fixed rate methods (Figure 7, left panel). At the trial pay

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

rates, it would cost \$7.23 million to map the entire continent, whereas it would cost just \$3.47 million using a variable pay rate, which is not much higher than \$3.04 million that would be required if pay was at the lower fixed rate. Applying the cheaper variable rate scheme, the cost would be just \$2.07 million to map all of Africa. Applying these alternate payments rates for the sites mapped during the trial reveals that variable rates would produce overall effective wages comparable to fixed rates, while paying nearly 50% higher, on average, for mapping sites with fields (Table 3).

Total mapping time estimates vary widely according to the number of workers involved, ranging from more than 19 years to map the whole continent with just 100 workers involved (i.e. 100 mapping hours per day) at the upper confidence limits of mapping time, to 1.2 - 3.8 years (mean = 1.9 years) if 500 workers map.

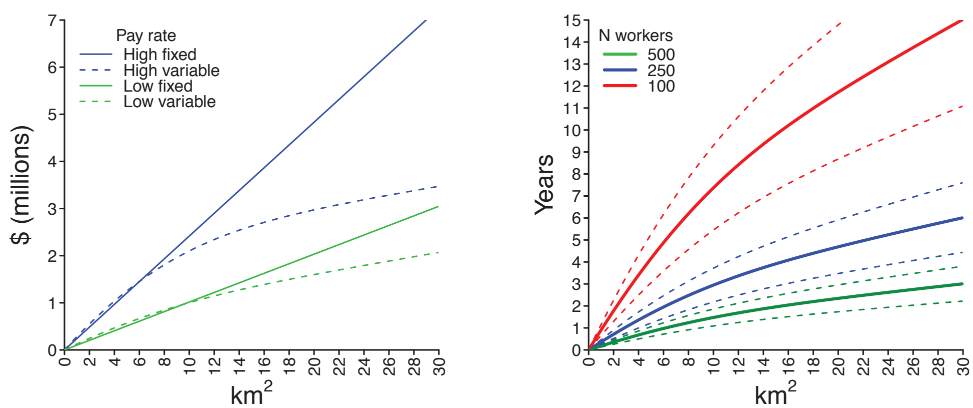

Figure 7: Several estimates of the cumulative cost (left) and time (right) of using DIYlandcover to map cropland throughout Africa. Cost estimates are based on either fixed (solid line) or variable (dashed line) rates, using higher (blue) and lower (green) cost models for each case. The cumulative mapping time is calculated in years, based on three different levels of worker involvement (100 [red], 250 [blue], and 100 [green] workers, each mapping for 1 hour/day). Solid lines indicate regression-predicted means, dashed lines the upper and lower confidence bounds for each model.

5. Discussion

The initial trial demonstrates that DIYlandcover can be an effective platform for generating high quality maps of discrete landcover types. This quality is attributable to humans' superior ability to recognize objects in patterns

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

Table 2: Effective wages for workers (in \$ hr^{-1}) under two fixed and two variable payment schemes, calculated as overall averages and for mapping sites with and without fields, and including mean bonus rates.

Payment method	Overall	With fields	Without fields
Fixed high	11.65	3.55	14.53
Variable high	10.16	5.19	12.16
Fixed low	4.44	1.38	5.59
Variable low	4.43	2.07	5.39

in noisy backgrounds, and is the reason why expert image interpretation is a key component of training and assessing existing landcover mapping algorithms (e.g. Fritz et al., 2011, 2012; Hansen et al., 2013). Here we found that workers with less than 24 hours of mapping experience were able to map cropland with 91% accuracy. Although accuracy and mapping precision decreased when sites contained crop fields, and in proportion to the complexity of those fields (Fig. 6), the overall accuracy was higher than the latest generation landcover dataset of comparable resolution (82%; Fritz et al., 2015), and was close to that achieved by GTI's trained workers. Compared to GTI's performance at sites with fields, using the most comparable accuracy metric (Equation 4), DIYlandcover showed similar performance–even though the score was 6% lower than GTI's 95% (see Appendix S1)–because GTI mapped using a more inclusive set of rules, thereby reducing error rates, and DIYlandcover's accuracy algorithms are more precise than the one used to assess GTI performance (see Fig. 6B and Appendix S1). The positive relationship we see between worker experience and score (Fig. 6F) also suggests that DIYlandcover's accuracy improves with time, and we expect that the implementation of bonus payments for performance will also improve mapping skill. These latter two points will need to be evaluated after a lengthier period of production, as does the affect of the different accuracy component weights (Eq. 1) in terms of influencing worker–and thus system–performance.

The trial also suggested that DIYlandcover has the potential to generate map data relatively rapidly, given an adequate number of workers (Figure 7). With 500 workers each contributing one hour of work per day, we estimate that all of Africa could potentially be mapped in approximately 1.9 years, which is roughly six month faster than the time needed to create GTI's map for South Africa. It is not unreasonable to think that this level of worker involvement is feasible on a for-pay crowdsourcing platform, particularly given

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

the payment rates we applied, which were substantially greater than the \$2 hr^{-1} received by Mechanical Turk workers (Marvit, 2014).

Our cost assessment (Fig. 7) indicates that linking pay to cover occurrence probabilities–and site selection to a finer gradation of weights–could greatly reduce the overall costs of "wall-to-wall" mapping, while maintaining fair worker wages. Although \$3 million is a significant amount of money, we argue that it would be a fairly cheap price to pay for a vector-based map of individual crop fields covering all of Africa. The overall mapping costs could be reduced to \$2 million if payments are made at the lower rates we assessed. We do not advise this approach when running the software on MT, as the worker base is primarily in the US given Amazon's fairly strict payment rules[3], and because there is growing concern about the exploitative nature of crowdsourcing (Marvit, 2014). However, it may be possible to pay *fairly* at lower rates if DIYlandcover is ported to job sites where workers can access it from countries with lower prevailing wages. For instance, in our example, had we been able to enlist workers in South Africa, where the exchange rate favors the dollar, we could have paid less and had mapping undertaken by workers who were familiar with those landscapes.

Other costs related to system development time could also be reduced, particularly with respect to generating reference maps. Our trial reference maps took several weeks to digitize, and were based on the judgement of a few people. A third type of mapping HIT, one that allows repeated mapping by multiple workers, would help mitigate these problems of cost and subjectivity. The resulting maps could be combined to create a more robust "truth" based on between-worker agreement, as illustrated by the combined maps from the eight qualification sites used in the trial (Fig. 7). This approach could greatly minimize the time required to develop reference data, and we suspect that the consensus maps of many workers (which could be weighted based on worker quality scores) will be more accurate than those of one or two experts (the last assumption must be verified against field-collected boundary data). Another advantage of this approach, which will be incorporated in the next update of DIYlandcover, is that the quality assessment protocol would essentially become a form of peer review.

Of course, the costs and necessary mapping time assessed here may still be too much for many users who need spatially comprehensive, large area land-

[3]www.mturk.com/mturk/help?helpPage=worker

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

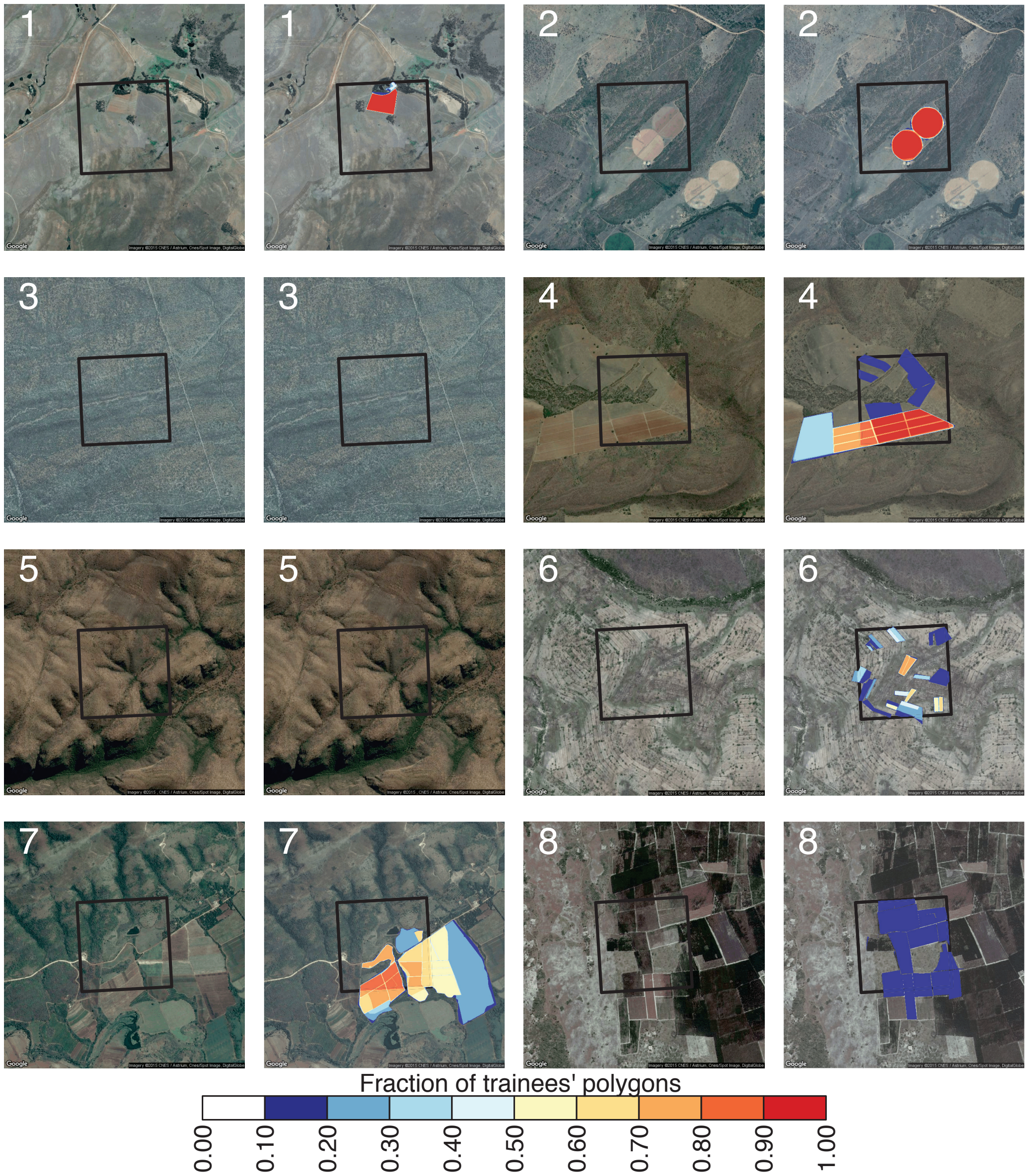

Figure 8: The eight sites used in the South Africa trial qualification test. Columns 1 and 3 show the unmapped imagery; columns 2 and 4 display the combined maps of all 19 trainees. Map colors indicate the fraction of trainee maps that overlap.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

cover maps. To reduce costs, DIYlandcover could be ported to a server that supports voluntary crowdsourcing efforts, similar to the Geo-Wiki project (Fritz et al., 2012), but this would not address the problem of longer mapping times. An alternative, more advanced, application would be to use DIYlandcover to train and test newer computer vision approaches for mapping noisy landcover types, such as smallholder crop fields (e.g. Debats et al., 2015). In this case, DIYlandcover would work iteratively with the algorithm until an acceptable level of accuracy is achieved, with site selection weighted towards areas of greatest classification error after each step. This approach could strike the best balance between cost, mapping speed, and accuracy, as it would harness the complementary strengths of human (more effective at recognizing patterns in noisy RGB or black and white images) and computer image classifiers (able to extract patterns in high dimensional data, such as multispectral imagery, which are hard for humans to interpret). An alternative possibility for this use case—where DIYlandcover validates broader-scale methods—would be test and refine the judgement-based size class estimates created under the Geo-Wiki project (Fritz et al., 2015). DIYlandcover is highly complementary to this methodology, given its emphasis on precisely measuring landcover geometries.

Beyond the questions of accuracy, cost, and time, the geometric detail captured by vector boundaries is a key data dimension that is lacking from current landcover products, and difficult to obtain from automated approaches. It is this capability to map individual features that may appeal to the broadest range of potential users, as geometry data provide information on a number of important social, economic, and environmental processes. For instance, crop field sizes can be effective predictors of agricultural economic metrics, and as such development-oriented agencies may be significantly interested in using this tool to generate these data (Fritz et al., 2015). Alternatively, conservationists wanting to identify habitat fragments to protect may be interested in more precise boundary data, as patch geometry correlate with extinction (Laurance et al., 2011) and thus conservation values. Other potential uses include mapping buildings, burn scars, water holes, and termite mounds, to name a few.

Acknowledgements

This work was supported by funds from the Princeton Environmental Institute Grand Challenges program, the NASA New Investigator Program

(NNX15AC64G), and the National Science Foundation (SES-1360463 and BCS-1026776).

Supplementary materials

Supplementary methods are presented in Appendix S1; Supporting figures S1 and S2 are found in AppendixS2; Appendix S2 contains the R code used to analyze trial results and create related figures.

References

Ahn, L. v., Blum, M., Hopper, N. J., Langford, J., 2003. CAPTCHA: Using Hard AI Problems for Security. In: Biham, E. (Ed.), Advances in Cryptology EUROCRYPT 2003. No. 2656 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 294–311.

Allahbakhsh, M., Benatallah, B., 2013. Quality control in crowdsourcing systems. IEEE Internet Computing 17 (2), 76–81.

Allouche, O., Tsoar, A., Kadmon, R., Dec. 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology 43 (6), 1223–1232.

Biederman, I., 1987. Recognition-by-components: a theory of human image understanding. Psychological review 94 (2), 115.

Bivand, R., Rundel, C., 2013. rgeos: interface to geometry engine-open source (GEOS). R package ver. 0.33.< http://cran. r-project. org/web/packages/rgeos/index. html.

Conway, J., Eddelbuettel, D., Nishiyama, T., Prayaga, S. K., Tiffin, N., 2012. RPostgreSQL: R interface to the PostgreSQL database system (2010). R package version 0.1-7.

Debats, S., Luo, D., Estes, L., Fuchs, T., Caylor, K., 2015. A generalized computer vision approach to mapping agricultural fields in Sub-Saharan Africa. PeerJ PrePrints 3, e1688.

Developers, G., 2012. Google Maps API.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1030v2 | CC-BY 4.0 Open Access | rec: 14 Nov 2015, publ: 14 Nov 2015

Estes, L. D., Beukes, H., Bradley, B. A., Debats, S. R., Oppenheimer, M., Ruane, A. C., Schulze, R., Tadross, M., Dec. 2013a. Projected climate impacts to South African maize and wheat production in 2055: a comparison of empirical and mechanistic modeling approaches. Global Change Biology 19 (12), 3762–3774.

Estes, L. D., Bradley, B. A., Beukes, H., Hole, D. G., Lau, M., Oppenheimer, M. G., Schulze, R., Tadross, M. A., Turner, W. R., Aug. 2013b. Comparing mechanistic and empirical model projections of crop suitability and productivity: implications for ecological forecasting. Global Ecology and Biogeography 22 (8), 1007–1018.

Estes, L. D., Paroz, L.-L., Bradley, B. A., Green, J. M., Hole, D. G., Holness, S., Ziv, G., Oppenheimer, M. G., Wilcove, D. S., Apr. 2014. Using changes in agricultural utility to quantify future climate-induced risk to conservation. Conservation Biology 28 (2), 427–437.

Flanagin, A. J., Metzger, M. J., Jul. 2008. The credibility of volunteered geographic information. GeoJournal 72 (3-4), 137–148.

Fraternali, P., Castelletti, A., Soncini-Sessa, R., Ruiz, C. V., Rizzoli, A. E., 2012. Putting humans in the loop: Social computing for Water Resources Management. Environmental Modelling & Software 37, 68–77.

Fritz, S., McCallum, I., Schill, C., Perger, C., Grillmayer, R., Achard, F., Kraxner, F., Obersteiner, M., Aug. 2009. Geo-Wiki.Org: The Use of Crowdsourcing to Improve Global Land Cover. Remote Sensing 1 (3), 345–354.

Fritz, S., McCallum, I., Schill, C., Perger, C., See, L., Schepaschenko, D., van der Velde, M., Kraxner, F., Obersteiner, M., May 2012. Geo-Wiki: An online platform for improving global land cover. Environmental Modelling & Software 31, 110–123.

Fritz, S., See, L., 2008. Identifying and quantifying uncertainty and spatial disagreement in the comparison of Global Land Cover for different applications. Global Change Biology 14 (5), 1057–1075.

Fritz, S., See, L., McCallum, I., You, L., Bun, A., Moltchanova, E., Duerauer, M., Albrecht, F., Schill, C., Perger, C., Havlik, P., Mosnier, A., Thornton, P., Wood-Sichra, U., Herrero, M., Becker-Reshef, I., Justice, C., Hansen, M., Gong, P., Abdel Aziz, S., Cipriani, A., Cumani, R., Cecchi, G., Conchedda, G., Ferreira, S., Gomez, A., Haffani, M., Kayitakire, F., Malanding, J., Mueller, R., Newby, T., Nonguierma, A., Olusegun, A., Ortner, S., Rajak, D. R., Rocha, J., Schepaschenko, D., Schepaschenko, M., Terekhov, A., Tiangwa, A., Vancutsem, C., Vintrou, E., Wenbin, W., van der Velde, M., Dunwoody, A., Kraxner, F., Obersteiner, M., 2015. Mapping global cropland and field size. Global Change Biology 21 (5), 1980–1992.

Fritz, S., See, L., Rembold, F., 2010. Comparison of global and regional land cover maps with statistical information for the agricultural domain in Africa. International Journal of Remote Sensing 31 (9), 2237–2256.

Fritz, S., You, L., Bun, A., See, L., McCallum, I., Schill, C., Perger, C., Liu, J., Hansen, M., Obersteiner, M., 2011. Cropland for sub-Saharan Africa: A synergistic approach using five land cover data sets. Geophysical Research Letters 38, L04404.

Frhlich, B., Bach, E., Walde, I., Hese, S., Schmullius, C., Denzler, J., 2013. Land cover classification of satellite images using contextual information. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 3, W1.

GeoTerraImage, 2008. South African crop field boundaries. Tech. rep., http://www.geoterraimage.com.

Gibbs, H. K., Ruesch, A. S., Achard, F., Clayton, M. K., Holmgren, P., Ramankutty, N., Foley, J. A., Sep. 2010. Tropical forests were the primary sources of new agricultural land in the 1980s and 1990s. Proceedings of the National Academy of Sciences 107 (38), 16732–16737.

Gross, D., Dubois, G., Pekel, J.-F., Mayaux, P., Holmgren, M., Prins, H., Rondinini, C., Boitani, L., Mar. 2013. Monitoring land cover changes in African protected areas in the 21st century. Ecological Informatics 14, 31–37.

Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S. A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S. J., Loveland, T. R., Kommareddy, A., Egorov, A., Chini, L., Justice, C. O., Townshend, J.

R. G., Nov. 2013. High-Resolution Global Maps of 21st-Century Forest Cover Change. Science 342 (6160), 850–853.

Hardy, M., Dziba, L., Kilian, W., Tolmay, J., 2011. Rainfed Farming Systems in South Africa. In: Tow, P., Cooper, I., Partridge, I., Birch, C. (Eds.), Rainfed Farming Systems. Springer Netherlands, pp. 395–432.

Howe, J., 2006. The rise of crowdsourcing. Wired magazine 14 (6), 1–4.

Jacobson, A., Dhanota, J., Godfrey, J., Jacobson, H., Rossman, Z., Stanish, A., Walker, H., Riggio, J., Oct. 2015. A novel approach to mapping land conversion using Google Earth with an application to East Africa. Environmental Modelling & Software 72, 1–9.

Jain, M., Mondal, P., DeFries, R. S., Small, C., Galford, G. L., Jul. 2013. Mapping cropping intensity of smallholder farms: A comparison of methods using multiple sensors. Remote Sensing of Environment 134, 210–223.

Laurance, W. F., Camargo, J. L., Luizo, R. C., Laurance, S. G., Pimm, S. L., Bruna, E. M., Stouffer, P. C., Bruce Williamson, G., Bentez-Malvido, J., Vasconcelos, H. L., Van Houtan, K. S., Zartman, C. E., Boyle, S. A., Didham, R. K., Andrade, A., Lovejoy, T. E., Jan. 2011. The fate of Amazonian forest fragments: A 32-year investigation. Biological Conservation 144 (1), 56–67.

Liang, X., Lettenmaier, D. P., Wood, E. F., Burges, S. J., 1994. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. Journal of Geophysical Research 99 (D7), 14415.

Licker, R., Johnston, M., Foley, J. A., Barford, C., Kucharik, C. J., Monfreda, C., Ramankutty, N., Nov. 2010. Mind the gap: how do climate and agricultural management explain the yield gap of croplands around the world? Global Ecology and Biogeography 19 (6), 769–782.

Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P., Vandenberg, J., Sep. 2008. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. Monthly Notices of the Royal Astronomical Society 389 (3), 1179–1189.

Lobell, D. B., Mar. 2013. The use of satellite data for crop yield gap analysis. Field Crops Research 143, 56–64.

Marvit, M. Z., Feb. 2014. How Crowdworkers Became the Ghosts in the Digital Machine. The Nation.

Monfreda, C., Ramankutty, N., Foley, J. A., 2008. Farming the planet: 2. Geographic distribution of crop areas, yields, physiological types, and net primary production in the year 2000. Global Biogeochemical Cycles 22, GB1022.

Ohori, K. A., Ledoux, H., Meijers, M., Oct. 2012. Validation and automatic repair of planar partitions using a constrained triangulation. Photogrammetrie - Fernerkundung - Geoinformation 2012 (5), 613–630.

Ozdarici-Ok, A., Akyurek, Z., Sep. 2014. Object-Based Classification of Multi-temporal Images for Agricultural Crop Mapping in Karacabey Plain, Turkey. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-7, 127–132.

Ozdarici-Ok, A., Ok, A. O., Schindler, K., May 2015. Mapping of Agricultural Crops from Single High-Resolution Multispectral ImagesData-Driven Smoothing vs. Parcel-Based Smoothing. Remote Sensing 7 (5), 5611–5638.

Ramankutty, N., Evan, A. T., Monfreda, C., Foley, J. A., Jan. 2008. Farming the planet: 1. Geographic distribution of global agricultural lands in the year 2000. Global Biogeochemical Cycles 22, 19 PP.

Ruesch, A., Gibbs, H. K., 2008. New IPCC Tier-1 global biomass carbon map for the year 2000. Carbon Dioxide Information Analysis Center (CDIAC), Oak Ridge National Laboratory, Oak Ridge, Tennessee. Available online at: http://cdiac. ornl. gov/epubs/ndp/global_carbon/carbon_documentation. html.

Rulli, M. C., Saviori, A., D'Odorico, P., Jan. 2013. Global land and water grabbing. Proceedings of the National Academy of Sciences 110 (3), 892–897.

Schellekens, J., Brolsma, R., Dahm, R., Donchyts, G., Winsemius, H., Nov. 2014. Rapid setup of hydrological and hydraulic models using OpenStreetMap and the SRTM derived digital elevation model. Environmental Modelling & Software 61, 98–105.

Schroff, F., Criminisi, A., Zisserman, A., 2008. Object class segmentation using random forests. British Machine Vision Association, pp. 54.1–54.10.

Searchinger, T. D., Estes, L., Thornton, P. K., Beringer, T., Notenbaert, A., Rubenstein, D., Heimlich, R., Licker, R., Herrero, M., May 2015. High carbon and biodiversity costs from converting Africa's wet savannahs to cropland. Nature Climate Change 5 (5), 481–486.

See, L., Fritz, S., You, L., Ramankutty, N., Herrero, M., Justice, C., Becker-Reshef, I., Thornton, P., Erb, K., Gong, P., Tang, H., van der Velde, M., Ericksen, P., McCallum, I., Kraxner, F., Obersteiner, M., Mar. 2015. Improved global cropland data as an essential ingredient for food security. Global Food Security 4, 37–45.

Services, A. W., 2012. Amazon Mechanical Turk.

Sullivan, B. L., Wood, C. L., Iliff, M. J., Bonney, R. E., Fink, D., Kelling, S., Oct. 2009. eBird: A citizen-based bird observation network in the biological sciences. Biological Conservation 142 (10), 2282–2292.

Tokarczyk, P., Wegner, J., Walk, S., Schindler, K., Jan. 2015. Features, Color Spaces, and Boosting: New Insights on Semantic Classification of Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing 53 (1), 280–295.

Yan, L., Roy, D. P., Mar. 2014. Automated crop field extraction from multi-temporal Web Enabled Landsat Data. Remote Sensing of Environment 144, 42–64.