Pest Management In Cotton Farms: An AI-System Case Study from the Global South

Aman Dalmia, Jerome White, Ankit Chaurasia, Vishal Agarwal, Rajesh Jain, Dhruvin Vora, Balasaheb Dhame, Raghu Dharmaraju, Rahul Panicker *†
Wadhwani Institute for Artificial Intelligence

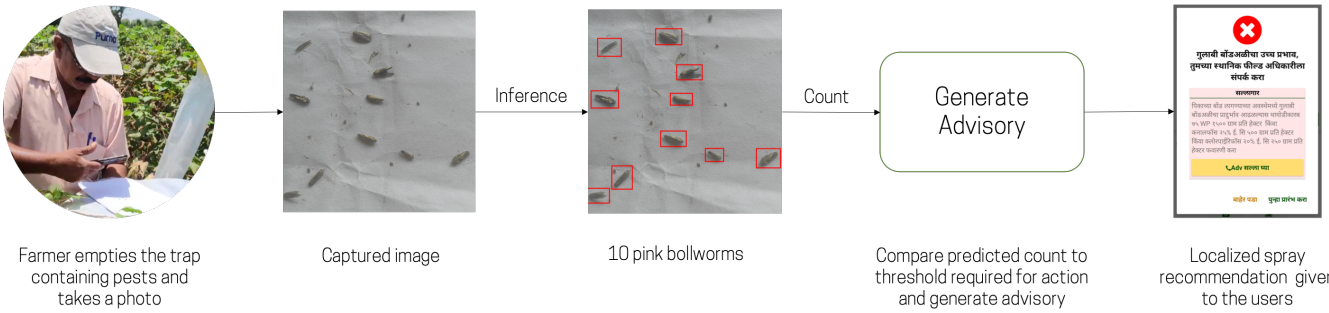


Figure 1: **Pesticide spray recommendation from pest trap images using AI.** We introduce an AI-based method for pest monitoring that is geared towards smallholder farmers. Using photos of pest traps in the field, the system provides a recommendation on whether to spray pesticide. This paper outlines that system along with the lessons learned during its deployment in the context of global development.

ABSTRACT

Nearly 100 million families across the world rely on cotton farming for their livelihood. Cotton is particularly vulnerable to pest attacks, leading to overuse of pesticides, lost income for farmers, and in some cases farmer suicides. We address this problem by presenting a new solution for pesticide management that uses deep learning, smartphone cameras, inexpensive pest traps, existing digital pipelines, and agricultural extension-worker programs. Although generic, the platform is specifically designed to assist smallholder farmers in the developing world. In addition to outlining the solution, we consider the set of unique constraints this context places on it: data diversity, annotation challenges, shortcomings with traditional evaluation metrics, computing on low-resource devices, and deployment through intermediaries. This paper summarizes key lessons learned while developing and deploying the proposed solution. Such lessons may be applicable to other teams interested in building AI solutions for global development.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; *Supervised learning by classification*; *Supervised learning by regression*; *Image representations*; **Multi-task learning**; **Neural networks**.

KEYWORDS

cotton farming, global development, object detection, artificial intelligence, deep learning, deployment lessons, pest monitoring

ACM Reference Format:
Aman Dalmia, Jerome White, Ankit Chaurasia, Vishal Agarwal, Rajesh Jain, Dhruvin Vora, Balasaheb Dhame, Raghu Dharmaraju, Rahul Panicker . 2020. Pest Management In Cotton Farms: An AI-System Case Study from the Global South . In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403363

1 INTRODUCTION

Nearly 100 million families across the world rely on cotton farming for their livelihood.[1] A large percentage of these families consist of smallholder farmers in developing countries. For these farmers, one of the most significant impediments to a successful season is mismanagement of pest infestation. Pest attacks not only cause crop loss, but can push farmers already fighting poverty into despair. After a pest attack in late 2017, for example, observers attributed a spike in farmer suicides in part to a widespread pest attack [16].

This paper describes an effort to address this problem in the context of challenges unique to the Global South, using deep learning, smartphone cameras, inexpensive pest traps, existing digital pipelines, and agricultural extension-worker programs. As outlined in Figure 1, our system uses an object-detection system trained to

*AD and JW contributed equally to this research.
†Corresponding author: aman@wadhwaniai.org

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403363

[1]Pesticide Action Network UK

identify and count pests caught in traps placed throughout a field. Using this count, it classifies the level of infestation and provides a recommendation on how to proceed. The system was developed and tested with farmers and farm extension workers throughout three cotton seasons across two different geographies in India.

What makes this problem unique are the challenges imposed by the settings in which this solution is intended to operate. The seasonal nature of cotton farming, for example, not only constrains data access, but contributes to its diversity. Pest occurrences in the field are markedly different from those in a lab setting. This in turn makes annotation difficult as it changes the landscape for domain experts responsible for establishing ground truth. Evaluation can also be a challenge when classical metrics used to judge machine learning models fail to completely align with what success means for the end-user. Finally, operating in a low-resource environment brings constraints around compute resources and network connectivity.

The contributions of this paper are twofold: **1)** introducing a new approach for trap-based pest management using deep learning, tailored toward real-world constraints common throughout the Global South, and **2)** outlining lessons learned from developing and deploying the approach for smallholder farmers. This work sits at the intersection of applied AI research and global development. We hope the insights here can be of use to other researchers looking to work in this emerging domain.

Section 2 begins by defining and motivating the problem, and detailing the related work. Section 3 motivates the solution by describing the unique challenges in solving it; from data collection, to algorithmic development, to deployment pathways. Section 4 details our approach to a few of the more technical challenges previously presented. Section 5 outlines some of the lessons learned throughout this process. Section 6 concludes the paper and offers future directions.

2 BACKGROUND

2.1 The Farmer's context

2.1.1 Pests: Both pink and American bollworms have been a long-standing nuisance for cotton farmers. Bollworms lay eggs that are difficult to detect and within a few days develop into larvae that can be difficult to manage. These larvae make their way into the cotton boll where they do irreparable damage to the crop.

2.1.2 Pesticides: Pesticides are a common intervention against bollworms. Knowing when to apply pesticides however, can be a difficult decision. Applying too soon can kill pests that are beneficial to a field's ecosystem, leaving it more susceptible to attack than it was before the spray. Spraying too late has the potential to be a wasted effort, as pest growth can exceed a level that is economically sensible to address. It is thus common to take a cautious approach and spray too frequently, something that is not only expensive, but environmentally hazardous, and potentially toxic [14].

2.1.3 Traps: Pheromone traps (Figure 2) are a tool that can be used as an early warning system to address this uncertainty. A lure within the trap attracts adult male bollworms during their mating period. The number of bollworms captured is then used as a proxy for larvae inception, and subsequently as an indicator for whether action is required to stunt larvae development. To be used correctly, traps must be emptied at regular intervals and their counts accurately interpreted as measures of infestation; knowing, for example, that finding a few pests is acceptable, but that finding a dozen is not.

Figure 2: **Description of a pheromone trap.** A pheromone trap is mounted to a post using its firm top. Bollworm moths enter the trap through the top, becoming trapped in the plastic bag that hangs below it. The contents can be emptied by untying the bottom of the bag. A pheromone lure hangs from the top (shown in red), which attracts the bollworms to the structure.

2.1.4 Extension workers: To assist with trap interpretation, farmers receive guidance from farm extension workers—representatives of larger programs with mandates to improve farmer livelihoods. Workers visit farms at a regular cadence, where part of their job is to observe and report trap catch. These observations are aggregated at a central authority, which then provides recommendations on whether to take action. This cycle is not only slow, but is susceptible to human error as identifying the correct pest class for an untrained person is a challenging task in itself.

2.1.5 The opportunity: It has been reported that proper intervention can increase a farmer's income by as much as 26 percent [29]. This has motivated several efforts at educating farmers on best practices around pest management. These systems, however, can be difficult for smallholder farmers to follow and maintain [15]. Instead, it is common for farmers to rely on external advice for guidance. We perceived an opportunity to support that guidance using AI, and to deploy such a solution at scale.

The problem was thus to provide accurate and localized assistance to farmers and extension workers in interpreting trap contents. What was encouraging was that the problem seemingly had an AI solution: contents of the trap essentially required recognition and counting. Equally as important was that the agricultural extension program had a preexisting network of farmers in the region; and that such extension programs are common in developing countries around the world where smallholder farming is the norm. Finally, the program had a working relationship with a software vendor, who had developed a smartphone application for basic farm-related data collection that was already in use by a wide range of farmers. Plugging into such a digital pipeline provides a natural pathway to scale. These three components—an AI solution, a programmatic partner, and the existence of a digital pipeline—motivated us to work on this problem.

2.2 Related work

2.2.1 AI-based pest detection. There have been several efforts to apply machine learning and artificial intelligence to pest detection.

Figure 3: **Example of our annotation.** Pink bollworms from a pheromone trap annotated with bounding boxes.

One class of approaches has focused on plant leaf characteristics for signs of damage [21, 23, 24] (see Prajapati et al. [22] for a review). Pest traps provide an opportunity to recognize infestation problems before signs of leaf damage arise; something particularly important for managing bollworms. There have been efforts focused on pheromone traps, however, these require special hardware to obtain images [6, 28] whereas our system can work on images captured simply from smartphone cameras.

Recent work that has applied object detection to pest recognition has utilized two-stage methods, empirically showing them to be more accurate than one-stage methods [9, 25, 26, 31]. However, they have also found them to be slower: in comparison to their best-performing two-stage system, Liu et al. [9] found that a one-stage approach decreased inference time by approximately 90 percent. Given our compute constraints, a one-stage approach was thus desirable. Other efforts that have proposed such approaches were Ding and Taylor [2], who developed an architecture based on LeNet [7], and Nam and Hung [20], who found SSD to be the best performing system in their context. While these efforts focus on the object detection algorithm, we propose an end-to-end framework based on multi-task learning where object detection is one of the tasks.

2.2.2 Deployment lessons. This paper follows a line of work detailing lessons learned from actual deployments of end-to-end machine learning systems. Haldar et al. [3], for example, describe the process by which an existing search platform was shifted to a deep learning model; Alibaba and eBay describe their techniques for implementing visual search (Zhang et al. [35] and Yang et al. [34], respectively); and Liu et al. [10] detail their approach to developing, deploying, and maintaining an employer name normalization task. These efforts are interesting not only for their technical contributions, but also because they have been deployed for real-world applications. However, these systems were not tailored to operate in the developing world, allowing many of the lessons presented in our work to be fresh perspectives.

There have been similar field study papers within the technology for development community [1, 18]. Although these efforts have not been AI-centered, our work has several shared experiences. Of particular note is the importance of on-ground partnerships for deployment.

2.2.3 AI for global development. Machine learning and AI have begun to find their way into the information and communication technologies and development (ICTD) community [5, 13, 30]. Much of the work in these areas has focused on controlled studies of usability, or proof of concepts, rather than long term deployments. In contrast, our solution was developed and deployed over three cotton seasons in two different geographies.

3 PROPOSED SOLUTION

Our approach to the problem was to use deep learning for identifying and counting pests emptied from the trap onto a sheet of paper. The count was then used to identify the infestation level and provide pesticide usage recommendations based on rules set by entomologists. These recommendations included: (1) no spray required; (2) no spray required now, but potentially required soon; and (3) spraying is strongly suggested. The solution was distributed through extension programs by plugging into existing technology platforms in their workflow.

3.1 Unique challenges

A key aspect of the proposed solution is the set of challenges that the context of the developing world imposes on it. We list them briefly below to contextualize the solution, then elaborate on some of them in Section 4.

Annotation: We relied on expert entomologists to assist with pest annotation. These experts were accustomed to high-resolution images of live insects. Our data consisted of pest images of much lower resolution, with many body parts withered off during emptying of the trap. This made it hard for them to correctly identify the pest class.

Data diversity: Our solution collects and analyzes data from a variety of users. Each user had different tendencies in how they empty the trap—how the pests are distributed, and how much extraneous material they include. Also, pests themselves exhibit different anatomies depending on their agro-climatic zones. These factors contribute to a diverse set of data.

Custom metrics: The foundation of our solution is object detection, where the common measure of model performance is average precision (AP) per class at a given bounding box overlap cutoff. Farmers, however, are not only impacted when our solution is right, but also when our solution is wrong. The relationship between AP and farmer benefit is therefore not straightforward, as false positives and false negatives are equally important.

Deployment through intermediaries: We worked with several partners in various aspects of the development and deployment process; namely farming extension programs and software partners who provided digital services to these extension programs. While these partners provide an avenue for deployment and scale, they also increase the cost of experimentation. Making system-level changes, even in sandboxes, has implications for their workflow and their reputation. These constraints also have ramifications for our system design. Things like the memory footprint of our models, for example, must be mindful of our software partner's overall footprint.

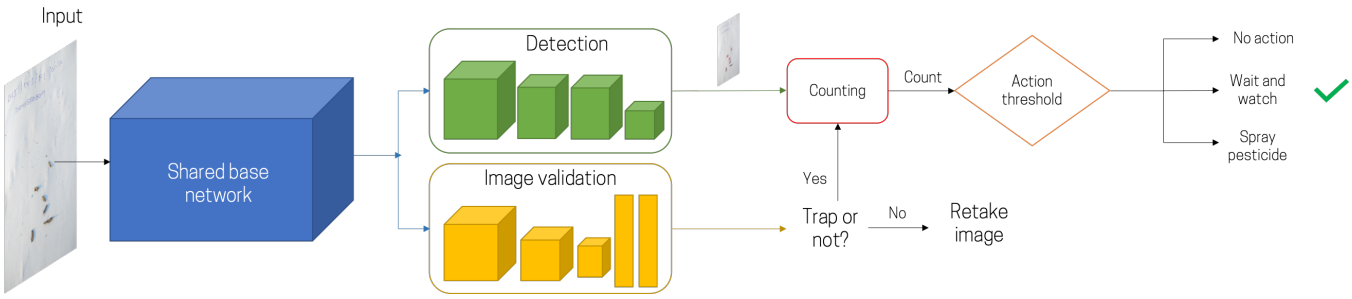


Figure 4: **Architectural framework.** An input image goes through the shared base network to produce a convolutional feature map that is fed into two branches: one for detection and one for image validation. If the validation branch predicts the image as non-trap, the user is asked to retake the photo; otherwise the detection head presents the bounding box predictions and the counts of the detected pests. Counts are used to make the final spray recommendation based on the action thresholds set by entomologists.

Table 1: **Data description.** Summary of our data collection effort, compartmentalized by phase and split. All data collected from Phase 3 was used as a test set.

		Phase 1		Phase 2		Phase 3	
Split	Class	Pics	Pests	Pics	Pests	Pics	Pests
Train	PBW	668	14222	232	111		
	ABW	1307	8600	217	277		
Val.	PBW	90	1978	23	21		
	ABW	157	1042	33	50		
Test	PBW	87	1642	28	18	660	2163
	ABW	160	1089	29	36		
Total		2469	28573	562	513	660	2163

Compute in low-resource settings: Many smallholder farms in India are located in remote areas, with inconsistent network coverage. This necessitates the ability to do as much on-device compute as possible, including model inference.

Seasonality and uncertainty: Cotton being a seasonal crop means that data collection and field evaluation have specific narrow windows of time in which they can be conducted. Additionally, whether there is a pest attack, and how strong such an attack may be, are highly variable. These factors mean development and programmatic timelines must be fluid and amenable to uncertainty.

Lack of digital literacy: Digital literacy levels vary widely, with multiple dimensions within that variance. Some users, for instance, are completely comfortable with certain interfaces, yet they are unaware of, or completely ignorant to, usage methods outside of that established pattern. This puts a premium on interface design and inference interpretability.

Lack of natural feedback loops: Financial metrics provide vital feedback for the deployment of AI solutions in commercial settings. Analogous feedback loops of similar reliability do not naturally exist in global development. For example, obtaining feedback on the impact of our solution on farmers requires an impact assessment mechanism to be established.

3.2 Data

For many tasks, there are either pre-existing datasets, or surrogate datasets sufficiently close to the expected distribution, on which initial models can be quickly developed. For our problem that was not the case. While several datasets have been presented in the academic community, those datasets are either proprietary [2, 12, 20, 32] or are too far from ground reality to be valuable in our setting [17, 31, 33]. We thus had to build a dataset from scratch. Such an effort should be seen as the norm, and not the exception, when building any machine learning solution for the developing world. This effort was made more difficult due to the seasonal nature of cotton farming. In India, cotton is commonly grown between September and January, the *kharif season*. Although it is also grown between March and May, the *summer season*, the number of regions in which this takes place is far less. Finally, irrespective of season, pest frequency is highly variable, limiting the number of opportunities to collect data and to conduct field experiments.

Data collection took place in several phases over three cotton seasons across two geographies for two pests types—pink (PBW) and American (ABW) bollworms. Table 1 presents a summary of this collection effort, factored by phase, pest class, and dataset split. Within this paper, models and subsequent results were derived from data within the highlighted cells. Specifically, models were trained using both bollworm types, but evaluation was performed only on pink bollworms. This decision was made in part to simplify presentation of our analysis. The collection process in each phase is described below.

3.2.1 Phase 1: Controlled data collection. The first phase of collection took place during the 2018 kharif season in Wardha (Maharashtra, India). A partner organization with expertise in cotton farming was engaged. This organization had a long-standing relationship with farmers in the area through their extension workers, as well as a cadence of visiting farms every alternate day. In addition to their existing field work, the workers were asked to maintain and take photos of pheromone trap contents by emptying the contents of the trap onto a plain white sheet of paper (Figure 3). They were then instructed to take a photo of the sheet using Open Data Kit (ODK) [4].

Figure 5: **Pest variety.** Variety in occurence of pink bollworms in our data, where features like scales, wings, and eyes of the same pest tend to wither off as they are emptied from the trap. The resolution of these pest images are also much lower compared to high resolution images used in previous studies and found on the internet, with which entomologists are more familiar. This poses a challenge not only for detection, but also for annotation.

3.2.2 Phase 2: Changing regions. The second phase of collection was conducted during the 2019 summer cotton season in Nannilam (Tamil Nadu, India). Unlike the previous season, data collectors were not tied to an organized extension program. Instead, they were identified and managed through a partnership with a lead farmer in the area. Data collectors were not farmers, but had basic knowledge of farming practices. Similar to Phase 1, ODK was used for image capture.

3.2.3 Phase 3: collection via deployment. The third phase of collection took place during the 2019 kharif season in Wardha. Rather than being a season devoted entirely to data collection, the objective was to test an early version of the solution and to do so with a wider audience. The extension workers who were a part of the outreach organization enlisted during Phase 1 served as the primary data collectors.

Across all phases, data collectors were asked not to engage the flash and to ensure bollworms were in focus. They were asked to remove extraneous objects like dirt and to avoid clumping by distributing the pests, but these requirements were not strictly enforced.

3.3 Model framework

3.3.1 Object detection. The majority of contemporary object detection systems can be broadly divided into two classes. Two-stage detectors first generate a series of proposals, then refine those proposals to come up with the final estimates. Single-stage detectors eschew the proposal generation step, directly predicting a fixed number of boxes in a single forward pass. In the context of pest recognition, single-stage detectors have been shown to reduce inference time by approximately 90 percent compared to most two-stage detectors [9]. Because such performance is paramount to the broader user experience, we decided to use a one-stage approach. Specifically, SSD was selected because it was well-established within the detection community, and was a good performer on pests [20].

3.3.2 Image validation. During data collection and field deployment it was common for users to submit images that did not contain trap-catch. It was clear that an end-to-end pipeline would also need image validation to reject outliers. To solve this problem we trained a standard image classification network [27] on a collection of non-bollworm images. This collection included images from the Common Objects in Context (COCO) dataset [8], along with leaf images and images of different types of pest traps collected during the three phases. The model was trained to provide a binary indication of whether the given image was of pheromone trap catch.

3.3.3 Multi-task learning. Because the image validation component and the SSD model utilized the same base network, there was an opportunity to combine the two architectures to simplify our deployment pipeline. This was accomplished by using the last layer of the base network in SSD and adding a new branch for image validation consisting of two convolutional layers, followed by an adaptive max-pool and fully-connected layers. The architecture of the image validation branch is similar to Simonyan and Zisserman [27]. During inference, the output of the detection branch is used only if the classification branch classifies the input image as a pheromone trap.

This network (Figure 4) was trained in a multi-task learning framework. The loss function was as follows:

$$L_{ssd} = L_{conf} + \alpha L_{loc} \tag{1}$$

$$L = L_{ssd} + \beta L_{validation} \tag{2}$$

where α and β were set to one. Equation 1 is the MultiBox loss, a combination of confidence (L_{conf}) and localization (L_{loc}) losses, introduced by Liu et al. [11]. $L_{validation}$ is the cross-entropy loss over two classes (trap versus non-trap).

3.3.4 Evaluation. The traditional way of evaluating object detection networks, average precision (AP) per class, was found to be suboptimal for our task (Section 4.3). As previously mentioned, we provide one of three recommendations to the user, only one of which involves spraying. We combine the other two into a single class and treat the task as a two-class problem. *Missed alarm rate* (MA) is the percentage of cases where the recommendation should have been to spray, but the system suggested otherwise. *False alarm rate* (FA) is the percentage of cases where no action should have been taken, but the system suggested spraying. Missed alarm rate and false alarm rate are equivalent to the false negative rate and the false positive rate, respectively. We chose to refer to them as MA and FA for communicating our results to relevant stakeholders who found false positives and false negatives difficult to quickly disentangle. Our system achieves 1.9 percent false alarm and 27.2 percent missed alarm rates.

3.4 Deployment

We initially developed an in-house mobile application that was distributed to extension workers and farmers through our agricultural program partners. We later integrated our application into that of a technology partner to the extension worker organization. The longstanding relationship between these two organizations provided us a clearer path to scale. System integration involved creating web services to communicate with our models. However, after compressing our models to work on the phone, integration through a software development kit (SDK) is also possible.

Table 2: **Ambiguity of optimal threshold across phases.** Performance of confidence thresholds optimized separately on Phase 1 and Phase 2 validation sets (rows) across all validation sets (columns).

		Phase 1		Phase 2		Phase 1+2	
	Conf.	MA	FA	MA	FA	MA	FA
Phase 1	0.2	0.00	5.41	-	8.70	0.00	6.67
Phase 2	0.7	49.06	0.00	-	0.00	49.06	0.00

4 ANALYSIS OF CHALLENGES

4.1 Annotation

4.1.1 Motivation. Image annotation involved drawing bounding boxes around the bollworms. Figure 3 provides an example. To gain first-hand knowledge of the problem domain, initial annotation was completed informally by the authors on a subset of images. This was done under periodic guidance from expert entomologists within the extension firm engaged in Phase 1 and 3. The difficulty of this process was that entomologists had little experience examining mobile phone images of post-catch bollworms. They were instead more accustomed to high-resolution images used more commonly in their previous studies. Our data (Figure 5) consists of pest images with much lower resolution, making it hard to correctly identify the pest class. Also, many of the features readily apparent on live or freshly caught bollworms had degraded in our set. Distinguishing features such as body scales, wings, and eyes were not discernible due to conditions within the trap and time to photograph. Finally, pests were often clumped together making it hard to establish clear boundaries between instances. Coupled with poor photo quality either due to human error or lighting conditions, these factors made accurate annotation a non-trivial task, even for experts.

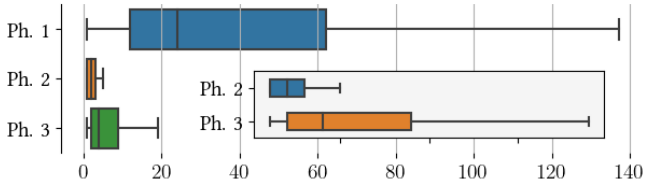
4.1.2 Solution. An annotation firm was later engaged that produced single sets of annotations per image. However, to address the problems just mentioned, we performed regular rounds of annotator agreement. Instead of producing a single set, the firm was asked to produce several independent sets of judgements per annotator. We would also participate by individually annotating the same set of images. We used the results not only to ascertain the quality of the firm, but to gauge the difficulty of the task itself by quantifying mismatch both internally and externally. Typically, annotator agreement is undertaken to sharpen an annotator set. Knowing the difficulty of the task for the experts, however, helped us maintain a more open attitude towards agreement performance.

4.2 Data diversity

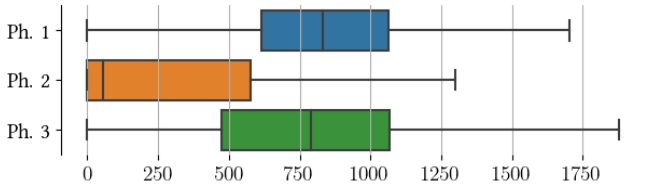
4.2.1 Motivation. Machine learning models have a better chance of performing well if the data captured during deployment comes from a similar distribution to that on which they were trained. This is a difficult property to maintain in our setting, where there are several sources of heterogeneity. Cotton fields are present in a variety of agro-climatic zones, each having an impact on the etymology of a pest. Due to the nature of human input, images also exhibit various spatial distributions of pests, as well as a variety of non-pest material present in the trap.

Table 3: **Estimating important correlates for relevant metrics.** Pearson correlation coefficients between aggregate measures and model performance. Values are derived using a box confidence threshold of 0.2. Highlighed values are significant at $p < 0.05$.

	AP		False Alarms		Correct Action	
Measure	Corr.	p-value	Corr.	p-value	Corr.	p-value
Count	0.35	0.00	-0.01	0.97	0.13	0.21
Size	-0.14	0.18	-0.01	0.93	-0.07	0.51
Spread	0.29	0.00	0.29	0.05	-0.14	0.15



(a) Number of pests per image. Inset is a zoom of the Phase 2 and 3 distributions.



(b) Pest density: median number of pixels between pests, per image

Figure 6: **Image-level characteristics.** Descriptive characteristics of the pest data set across phases. Figure 6a plots the PDFs of the count per image for all phases. Although the distribution for Phase 3 might appear to seem close to Phase 2, on closer inspection, it is clear that that is not the case. Figure 6b indicates that the PDF of the density for Phase 3 matches that of Phase 1 closely.

Given that Phase 3 was deployment focused, there was a requirement to decide the best-performing model before its start. To do this we trained a model on the combined Phase 1 and Phase 2 training sets. For validation, we used the Phase 1 and Phase 2 sets separately and in conjunction via a third set consisting of their union. The optimal confidence threshold for each phase differed to the extent that there was no clear choice as to which was the most informative candidate (Table 2). This section focuses on the heterogeneity in our data as a means of reasoning about this mismatch. Using data from Phase 1 and 2 in comparison to, and in conjunction with, data from Phase 3, we discern characteristics of the distributions of each dataset and ascertain their importance for future deployments.

4.2.2 Solution. Our approach involved analyzing measurements derived from annotated bounding boxes of pests; specifically, pest count, pest size, and pest density. *Pest size* is the area of a bounding box relative to that of the parent image. It is aggregated per image by taking the mean across pest sizes for that image. *Pest density* per image is the median distance between the center coordinates of all pests in that image.

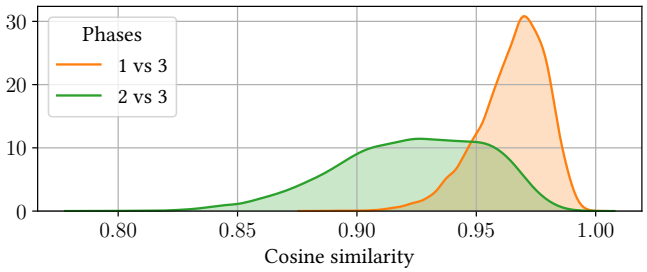


Figure 7: **Pest-level characteristics.** Probability density functions of the cosine similarity between latent representations of pests in Phase 3 with pests in Phase 1 and 2.

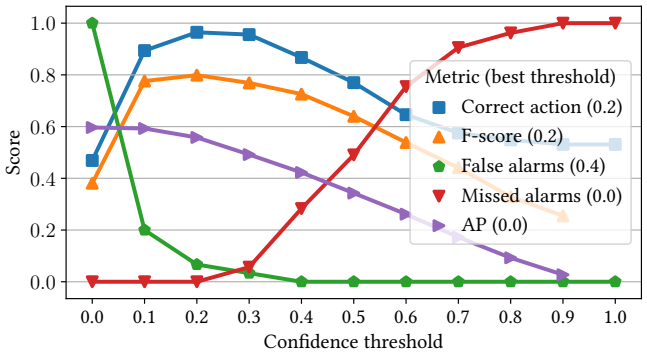


Figure 8: **Effect of metrics on optimal threshold.** Change in performance on the validation set across various metrics and box confidence thresholds. Optimizing for AP, the commonly used metric in object detection, can give a very different threshold compared to optimizing for correct action.

We first understand the relationship between these measures and model performance using a combined Phase 1 and 2 validation set. Table 3 presents Pearson correlation coefficients between image annotation metrics, and model performance scores. Significant correlations were found between AP and the measurements of count and density. False alarms were also found to be significantly correlated to density. Keeping these specific relationships in mind, we bring in Phase 3 data and study how the measurement distributions compare. Figure 6 presents a visualization. The pest count of Phase 3 seems closer to that of Phase 2 than to Phase 1. Detailed examination however reveals that the distribution is different. Phase 3 pest density not only resembles that found in Phase 1, but it does so quite closely. Combined with our knowledge of measurement importance, had these distributional relationships been identified earlier we would have had more faith in our decision to use the Phase 1 confidence threshold during the season.

Whereas annotation characteristics are derived from the bounding box, image characteristics are derived from the actual image contained within that bounding box. Each pest is first cropped from its image using its bounding box coordinates. Crops are then size-normalized and used to train a fully connected autoencoder to learn a latent representation. The network was trained using crops from Phase 1 and 2. Once complete, it was used to calculate representations of crops from Phase 3. The cosine similarity was taken between each representation from Phase 3 and representations from Phase 1 and Phase 2, respectively. The maximum similarity score was then kept for each image pair. The result was thus two maximum scores, one with respect to Phase 1 and another with respect to Phase 2, for each image in Phase 3. Figure 7 visualizes the two probability density functions. On average, pests in Phase 3 were more similar to those in Phase 1 (mean similarity 0.97) than they were to those in Phase 2 (mean similarity 0.92). A t-test was conducted to compare these distributions and their difference was found to be significant, $t(12724.8) = 116.88, p < 1e-99$.

4.3 Choice of evaluation metrics

4.3.1 Motivation. Because smallholder farmers are often cash sensitive, false positives and false negatives from a decision support system can have an outsized impact on their livelihood. In this context, a false positive means recommending a pesticide intervention that may not be necessary, leading to over-spraying. A false negative means a farmer may not take action when they actually should.

Table 4: **Test performance across thresholds.** Performance on Phase 3 test data using confidence thresholds proposed during validation.

	False Alarms (N=627)		Missed Alarms (N=33)	
Thresh.	Mean	SD	Mean	SD
0.0	98.4%	0.13	0.0%	0.00
0.2	1.9%	0.14	27.2%	0.45
0.4	0.2%	0.04	66.7%	0.47

Deciding to move a model into production after only evaluating it with traditional performance metrics runs the risk of being disconnected from ground reality. The common measure of model performance in object detection, the foundation of our solution, is average precision (AP) per class at a given bounding box overlap cutoff; AP at IOU 0.5, for example, where IOU is the intersection of two boxes divided by their union. The relationship between this measurement and the ultimate farmer utility is not one-to-one. This section quantifies that phenomenon and outlines relevant alternatives.

4.3.2 Solution. The detection head described in Section 3.3 outputs 200 box predictions for each class. A confidence threshold is applied on the predictions for each class to return the final detections. To evaluate the difference between metrics, we look at optimal model thresholds based on the performance of our validation set using various metrics. Along with AP, missed alarm rate, and false alarm rate, we introduce two additional measurements: mean absolute error (MAE), which is the absolute value of the difference between the inferred and actual counts; and *correct action*, which combines MA and FA, being true when either FA or MA are false.

A model was trained using the combined training set from Phase 1 and 2 (Table 1). The confidence threshold was then iteratively changed from 0 to 1 in steps of 0.1. At each step, the model

was evaluated using the combined validation set from the same two phases. Figure 8 plots this iterative evaluation for the metrics outlined. From this experiment, a model based on the performance of AP would use a threshold value of zero, while other metrics suggest otherwise. Table 4 presents how these thresholds perform on Phase 3 test data focusing on the farmer-centric evaluation metrics. By definition, an image can only be in one of the two alarm categories. Thus, the means presented in Table 4 are over images in their respective categories, where category size is denoted "N". While the AP inspired threshold minimized the propensity of false alarms, it was unable to mitigate missed alarms.

4.4 Compute in low-resource settings

4.4.1 Motivation. Compute resources are largely constrained by two factors. The first is mobile network coverage. Many smallholder farms in India are located in areas that are poorly connected. Relying too heavily on upload or download capability in these conditions can hamper user experience. The second constraint comes from integrating products into partner interfaces. Solutions need to be mindful of their existing size and quality of service guarantees. Evidence of these challenges was seen during our Phase 2 data collection effort, during which part of the season was used to test the end-to-end application. Our solution was placed on 17 phones that belonged to a mix of extension workers and lead farmers. The experiment ran for 60 days across 25 farms, providing 89 recommendations based on pink bollworm catch. The average per image upload time was approximately 38.5 seconds (95% CI [33.6, 43.3]), which represented 48 percent of task completion time. For users, this was especially problematic in cases where they had network connectivity in a location different from where they took the photo. In these conditions, an additional travel component was required to complete the task. This inconvenience was exacerbated in cases where a retake was required: a farmer would take a photo in the field, walk to an area with connectivity to perform the upload, then be forced to walk back into the field for a retake upon learning the image was invalid. In addition to the log analysis, qualitative follow-ups showed that these upload times would be a barrier to adoption.

4.4.2 Solution. To get compute closer to the field and mitigate network usage, we ported our multi-task network onto the phone. We first established a target model size by examining the size of our partner's app, and by understanding the general sizes of other agri-tech apps—between 15 and 20 MB generally. We decided that a model size of 5MB would be a good target. Getting to this size was essentially an exercise in model compression. We adopted a version of filter-level pruning known as iterative pruning, specifically using the technique developed by Molchanov et al. [19]. The idea was to iteratively prune a fixed number of filters followed by training the pruned model. Molchanov et al. [19] introduced this technique for image classification networks. We adapted this technique for our multi-task model, which consisted of both a detection head and a classification head. We pruned 1024 filters in each pruning iteration, followed by 30 epochs of training, until 80 percent of the total number of filters in the original model were pruned. The weights of the final pruned model were saved in half-precision to further reduce the memory footprint on disk.

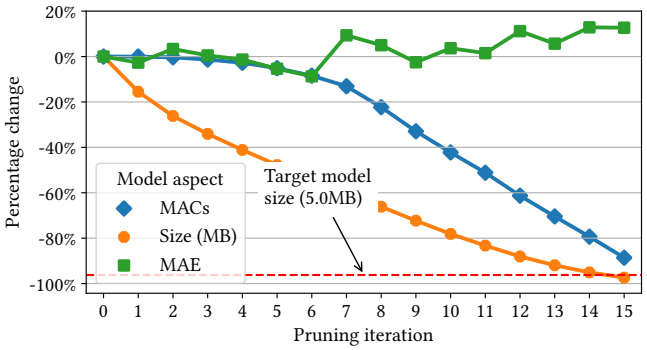


Figure 9: **Pruning progress.** Change in the memory footprint and the number of multiply accumulate operations (MACs) of the compressed model, along with a corresponding performance metric (MAE), during the lifecycle of iterative pruning.

Figure 9 details this effort. Iteration zero represents the original, unmodified model described in Section 3.3; its size after quantization was approximately 132 MB. Subsequent iterations came after a round of pruning. The required model size was reached after 15 iterations, where the model performance (MAE) went from 0.91 to 1.02. In addition to a reduction in size, the compressed model was also less compute-intensive, as seen from the drop in multiply-accumulate operations (MACs).

5 LESSONS LEARNED

Train the expert: Annotation is a critical component of building an AI-based solution. In the context of the developing world, it is not unlikely that even domain experts may not be able to provide accurate annotations because of lack of exposure to the reality on the field. This can therefore lead to a two-way learning process where the expert might themselves need additional training.

Sample before deployment: As we continue to deploy our solution to new regions, we anticipate needing to make model decisions with incomplete information. Specifically, diversity in data is always going to be a factor whenever scale is concerned. One of the most common ways of dealing with this is by simply deploying the system and relying on online training to refine the model. This can have negative consequences if the test distribution is different from the train distribution, something that is problematic in cost-sensitive settings such as ours. Therefore, especially during the early stage of a project, an alternative is to collect a sample of the data before deployment, to understand which subset of training data matches the expected distribution, and take decisions accordingly.

Customize metrics: Although there might be well-established evaluation metrics for a given machine learning method being used to address a given problem, it does not necessarily imply that those metrics capture the end goal of the solution. It is important to keep this in mind while deciding which metrics to optimize. Where appropriate, custom metrics that do capture user needs may be required.

Infer offline: When building tools for the developing world, or even many rural settings, network connectivity should not be taken for granted. Our approach has been to do as much compute

offline as possible by compressing our model to a suitable size for a phone. It is important to quantify that need early on, setting the right expectations with stakeholders regarding latency, model size, and accuracy. Offline inference also allows better management of data privacy as the solution scales because there is finer control of what user data leaves the phone.

6 CONCLUSION

This paper presents a new technical approach for trap-based pest management using AI. The solution requires only a mobile phone camera, readily available pest traps, and works in areas without connectivity. It lays the groundwork for providing accessible, immediate, and actionable advice to farmers. The lessons outlined in this paper, a result of several seasons of on-ground experience, shed light on what can be expected when building AI-based solutions for the developing world.

During our deployments, we have found that users not only want suggestions, but also want to understand why those suggestions were made. Future work includes exploring explainability techniques to address this concern. Bollworms are just one of dozens of pests that are active on cotton farms. Immediate future work is to replicate parts of this effort to address those. Further, pest count is one of a myriad of factors that can be considered for spray decisions. Being able to take those factors into account simultaneously may strengthen our predictions. Finally, a controlled study would allow us to put this work into the broader farming context.

7 ACKNOWLEDGEMENTS

We would like to thank Raghavendra Udupa for his guidance during early phases of this project; the Welspun Foundation and V. Ravichandran for assistance with on-ground support; Rameshwar Bhaskaran and Siddarth Bhatia for their technical contributions; and Alpan Raval for his advice throughout. We would also like to thank the anonymous reviewers for their insightful feedback. A portion of this work was supported by a grant from the Tides Foundation under the Google AI Impact Challenge (1904-57755).

REFERENCES

[1] A. Cross, N. Gupta, and et al. 2019. 99DOTS: A Low-Cost Approach to Monitoring and Improving Medication Adherence. In *Intl. Conf. on Information and Comm. Technologies and Development*. ACM, Article 15, 12 pages.

[2] W. Ding and G. Taylor. 2016. Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture* 123 (2016), 17–28.

[3] M. Haldar, M. Abdool, and et al. 2019. Applying Deep Learning to Airbnb Search. In *Intl. Conf. on Knowledge Discovery & Data Mining*. ACM, 1927—1935.

[4] C. Hartung, A. Lerer, and et al. 2010. Open Data Kit: Tools to Build Information Services for Developing Regions. In *Intl. Conf. on Information and Comm. Technologies and Development*. ACM, 18:1–18:12.

[5] M. Jain, P. Kumar, I. Bhansali, and et al. 2018. FarmChat: A Conversational Agent to Answer Farmer Queries. *Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4, Article 170 (December 2018), 22 pages.

[6] R. Kalamatianos, I. Karydis, D. Doukakis, and M. Avlonitis. 2018. DIRT: The Dacus Image Recognition Toolkit. *Journal of Imaging* 4, 11 (2018).

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (Nov 1998), 2278–2324.

[8] T. Lin, M. Maire, S. Belongie, and et al. 2014. Microsoft coco: Common objects in context. In *European conf. on computer vision*. Springer, 740–755.

[9] L. Liu, R. Wang, C. Xie, and et al. 2019. PestNet: An End-to-End Deep Learning Approach for Large-Scale Multi-Class Pest Detection and Classification. *IEEE Access* 7 (2019), 45301–45312.

[10] Q. Liu, J. Chao, T. Mahoney, and et al. 2018. Lessons Learned from Developing and Deploying a Large-Scale Employer Name Normalization System for Online Recruitment. In *Intl. Conf. on Knowledge Discovery & Data Mining*. ACM, 556—565.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. 2016. SSD: Single shot multibox detector. In *European Conf. on computer vision*. Springer, 21–37.

[12] Z. Liu, J. Gao, G. Yang, H. Zhang, and Y. He. 2016. Localization and Classification of Paddy Field Pests using a Saliency Map and Deep Convolutional Neural Network. *Scientific Reports* 6, 20410 (2016).

[13] W. Ma, K. Nowocin, N. Marathe, and G. Chen. 2019. An Interpretable Produce Price Forecasting System for Small and Marginal Farmers in India Using Collaborative Filtering and Adaptive Nearest Neighbors. In *Intl. Conf. on Information and Comm. Technologies and Development*. ACM.

[14] F. Mancini, J. Jiggins, and M. O'Malley. 2009. Reducing the Incidence of Acute Pesticide Poisoning by Educating Farmers on Integrated Pest Management in South India. *Intl. J. of Occupational and Environmental Health* 15, 2 (2009), 143–151.

[15] F. Mancini, A. van Bruggen, and J. Jiggins. 2007. Evaluating cotton integrated pest management (IPM) farmer field school outcomes using the sustainable livelihoods approach in India. *Experimental Agriculture* 43, 1 (2007), 97–112.

[16] Meena Menon. 2018. The pink bollworm menace adds to Maharashtra cotton farmers distress. *Scroll* (August 2018).

[17] E. Mique and T. Palaoag. 2018. Rice Pest and Disease Detection Using Convolutional Neural Network. In *Proceedings of the 2018 Intl. Conf. on Information Science and System*. ACM, 147–151.

[18] A. Moitra, V. Das, and et al. 2016. Design Lessons from Creating a Mobile-Based Community Media Platform in Rural India. In *Intl. Conf. on Information and Comm. Technologies and Development*. ACM, Article 14, 11 pages.

[19] P. Molchanov, S. Tyree, T. Karras, and et al. 2017. Pruning convolutional neural networks for resource efficient inference. In *Conf. on Learning Representations*.

[20] N. Nam and P. Hung. 2018. Pest Detection on Traps Using Deep Convolutional Neural Networks. In *Proceedings of the 2018 Intl. Conf. on Control and Computer Vision*. ACM, 33–38.

[21] A. Parikh, M. Raval, C. Parmar, and S. Chaudhary. 2016. Disease Detection and Severity Estimation in Cotton Plant from Unconstrained Images. In *Intl. Conf. on Data Science and Advanced Analytics*. IEEE, 594–601.

[22] B. Prajapati, V. Dabhi, and H. Prajapati. 2016. A survey on detection and classification of cotton leaf diseases. In *Intl. Conf. on Electrical, Electronics, and Optimization Techniques*. 2499–2506.

[23] P. Revathi and M. Hemalatha. 2012. Advance computing enrichment evaluation of cotton leaf spot disease detection using Image Edge detection. In *Intl. Conf. on Computing, Comm. and Networking Technologies*. 1–5.

[24] A. Sarangdhar and V. Pawar. 2017. Machine learning regression technique for cotton leaf disease detection and controlling using IoT. In *Intl. Conf. of Electronics, Comm. and Aerospace Technology*, Vol. 2. 449–454.

[25] M. Selvaraj, A. Vergara, H. Ruiz, and et al. 2019. AI-powered banana diseases and pest detection. *Plant Methods* 15, 92 (2019).

[26] Y. Shen, H. Zhou, J. Li, F. Jian, and D. Jayas. 2018. Detection of stored-grain insects using deep learning. *Computers and Electronics in AgricultureOA* 145 (2018), 319–325.

[27] K. Simonyan and A. Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Intl. Conf. on Learning Representations*.

[28] Y. Sun, X. Liu, M. Yuan, L. Ren, J. Wang, and Z. Chen. 2018. Automatic in-trap pest detection using deep learning for pheromone-based Dendroctonus valens monitoring. *Biosystems Engineering* 176 (2018), 140–150.

[29] TechnoServe. 2019. *Towards Doubling Cotton Farmer Income*. Technical Report. The Sustainable Trade Initiative.

[30] C. Valderrama, F. Marzbanrad, L. Stroux, and et al. 2018. Improving the Quality of Point of Care Diagnostics with Real-Time Machine Learning in Low Literacy LMIC Settings. In *Conf. on Computing and Sustainable Societies*. ACM, Article 2, 11 pages.

[31] X. Wu, C. Zhan, Y. Lai, M. Cheng, and J. Yang. 2019. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. In *Conf. on Computer Vision and Pattern Recognition*. IEEE, 8787–8796.

[32] C. Xie, R. Wang, J. Zhang, and et al. 2018. Multi-level learning features for automatic classification of field crop pests. *Computers and Electronics in Agriculture* 152 (2018), 233–241.

[33] C. Xie, J. Zhang, R. Li, J. Li, P. Hong, J. Xia, and P. Chen. 2015. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture* 119 (2015), 123–132.

[34] F. Yang, A. Kale, Y. Bubnov, and et al. 2017. Visual Search at eBay. In *Intl. Conf. on Knowledge Discovery and Data Mining*. ACM, 2101—2110.

[35] Y. Zhang, P. Pan, Y. Zheng, and et al. 2018. Visual Search at Alibaba. In *Intl. Conf. on Knowledge Discovery & Data Mining*. ACM, 993—1001.