Human-Machine Collaboration for Fast Land Cover Mapping

Caleb Robinson,[1,*] **Anthony Ortiz,**[2,*] **Kolya Malkin,**[3] **Blake Elias,**[6] **Andi Peng,**[6]
Dan Morris,[4] **Bistra Dilkina,**[5] **Nebojsa Jojic**[6,†]
[1]Georgia Institute of Technology, [2]University of Texas at El Paso, [3]Yale University,
[4]Microsoft AI for Earth, [5]University of Southern California, [6]Microsoft Research

Abstract

We propose incorporating human labelers in a model fine-tuning system that provides immediate user feedback. In our framework, human labelers can interactively query model predictions on unlabeled data, choose which data to label, and see the resulting effect on the model's predictions. This bi-directional feedback loop allows humans to learn how the model responds to new data. We implement this framework for fine-tuning high-resolution land cover segmentation models and compare human-selected points to points selected using standard active learning methods. Specifically, we fine-tune a deep neural network – trained to segment high-resolution aerial imagery into different land cover classes in Maryland, USA – to a new spatial area in New York, USA using both our human-in-the-loop method and traditional active learning methods. The tight loop in our proposed system turns the algorithm and the human operator into a hybrid system that can produce land cover maps of large areas more efficiently than the traditional workflows. Our framework has applications in machine learning settings where there is a practically limitless supply of unlabeled data, of which only a small fraction can feasibly be labeled through human efforts, such as geospatial and medical image-based applications.

1 Introduction

Machine learning models are usually imagined as artificially "intelligent" agents that mimic human autonomy and generalization abilities: having explored their training environment, machine learning models are supposed to choose their actions independently and reliably in similar situations. While this notion of intelligence guides the design and testing of new algorithmic ideas, in practice, the resulting algorithms are rarely capable of either autonomy or generalization. Instead, human decision-making is present throughout a AI model's development and lifetime: researchers and engineers acquire data with a specific goal in mind, then work on finding and tuning the methods that handle the peculiarities of the data well. When the algorithm is eventually deployed, it often suffers from *domain shift*, where slight changes in the statistics of real-world input compared to the training input can degrade performance considerably. Thus, the algorithm is constantly reevaluated through human monitoring, which may trigger a process requiring repeated data acquisition and retraining (Sculley et al. 2015). Hence, most practical deployments are better thought of as examples of hybrid – rather than purely artificial – intelligence. Active learning loops can be seen as an approximate model of such hybrid human-machine intelligence, as long as humans are allowed deeper involvement than just as labeling oracles. More specifically, the hypothesis is that if humans are allowed to choose which samples to label, and subsequently *fine-tune* a deployed model with, then they will be able to correct model errors, such as those from input *domain shift*.

Image segmentation is an ideal task to test hybrid human-machine intelligence, as segmentation is a natural ability of humans (Griffiths, Abbott, and Hsu 2016) and one where humans can exploit the spatial structure of input to identify errors. Recent work has probed the complementary abilities of humans and machines on image labeling tasks (Cai et al. 2019; Nushi, Kamar, and Horvitz 2018). We investigate whether it is possible to maximize performance on one such application, land cover mapping from high-resolution satellite imagery, by directly integrating humans into the training loop instead of isolating the artificially intelligent component. Our methods can be applied in settings where the human-in-the-loop can quickly search and evaluate the deployed model over unlabeled examples. This is the case in geospatial image labeling tasks and medical image segmentation tasks (e.g., segmenting tumor-infiltrated lymphocytes in pathology imagery), where unlabeled points have a strong spatial structure (i.e., points can be thought of as part of a large continuous image).

We summarize our main contributions as follows:

- We design an *interactive web tool* that enables users to test a high-resolution land cover model on any patch of land on a satellite map, then – in the same interface – relabel pixels of their choosing and retrain ("fine-tune") the model in real time (see Fig. 2).
- We study the effectiveness of the *combination of different active learning query methods with different model fine-tuning methods* in an offline study and find that query-

[*]Work done while interning at Microsoft Research
[†]jojic@microsoft.com

Figure 1: National Agriculture Imagery Program (NAIP) aerial imagery (**top row**) with modeled land cover estimates (**bottom row**). Existing supervised learning models, trained for generating land cover labels from aerial imagery, do not generalize well due to the large spatial and temporal variances in aerial imagery. Creating accurate land cover maps at a massive scale therefore requires additional human interventions. We propose an interactive model fine-tuning system, coupling human labelers and machine learning models, for facilitating these interventions.

ing for labels at randomly selected points outperforms or nearly matches standard active learning *query methods* (see Fig. 3).

- In an online user study, we examine **how well human labelers function as sample *query methods* compared to automatic selection methods**. We find that humans perform significantly better, even compared to learning systems in which the model is told on which points it is making labeling errors (see Fig. 4).
- Furthermore, we show that **the value of human-provided labels increases with the time humans spend using the tool**.

2 Background

Active learning

A traditional active learning setup consists of a parameterized model, an unlabeled 'pool' of data, a data *query method* (also known as the *data selection method*), and a *labeling oracle*. One iteration of fine-tuning the model consists of 1.) utilizing the *query method* to choose data points for labeling, 2.) querying a *labeling oracle* for the labels, and 3.) fine-tuning the model parameters to these additional data samples (Settles 2012).

The purpose of the *query method* is to pick unlabeled data that, when labeled, will provide the largest benefit to the model. In active learning, the learner is allowed to ask for help by querying the label oracle, but it must know which samples to request labels for. Conventional approaches ask the oracle to label instances with low prediction confidence (Zhang 2017; Settles 2012), or consider the similarity between an unlabeled sample and existing labeled samples as a selection criterion (Zhang et al. 2018). Another recent approach models uncertainty in labeling oracles to improve the efficiency of active learning (Huang et al. 2016). Meta-learning (or "learning to learn") active learning query methods rely on existing labeled datasets drawn from the same distribution as the unlabeled data pool (Hsu and Lin 2015; Bachman, Sordoni, and Trischler 2017), and as such will not be effective when the model must be adapted to work in a shifted distribution. Finally, *query method* algorithms suffer from "unknown unknowns": a model's self-inspection does not reliably reveal what it does not model well. This is the case in most ML algorithms, including deep neural networks (Nalisnick et al. 2019).

On the other hand, by observing the effects of their decisions on a model being retrained on-the-fly, human labelers can adapt their own data selection process to reflect not only their understanding of the data, but also their developing intuition regarding the inner workings of the model and its adaptation algorithms.

Land cover mapping

Land cover mapping – the segmentation of aerial or satellite imagery into land cover classes such as "water", "forest", "field/low vegetation", or "impervious surface" (Fig. 1) – has attracted reinvigorated interest in machine learning research (Robinson et al. 2019; Demir et al. 2018; Rakhlin et al. 2018; Tian, Li, and Shi 2018; Kuo et al. 2018). High-resolution land cover maps are an essential component in environmental science, agriculture, forestry (Hansen et al. 2013), urban development (Zhang et al. 2013), the insurance and banking industries, and for demography in developing countries (Facebook 2019). Satellite imagery is being produced on an increasingly frequent basis. However, despite their importance, high-resolution land cover maps are not yet widely available as neither ML algorithms nor human labor scale appropriately (Robinson et al. 2019).

To a machine learning or computer vision researcher, land cover mapping is a semantic segmentation problem. Machine learning models are not yet able to generate high-resolution (1m / pixel) land cover labels with performance that matches human labeling. A major obstacle is that high-resolution land cover labels for training such models only

exist in small, specialized locations (Demir et al. 2018; Rakhlin et al. 2018; Yang and Newsam 2010; Castelluccio et al. 2015). In (Robinson et al. 2019), it is shown that a state-of-the-art deep neural network trained on 1m-resolution images and labels from a much larger (160,000 km^2) dataset (Chesapeake Conservancy 2017) in the Chesapeake Bay watershed (north-eastern US) still does not perform well in the mid-western US. Other recent work also utilizes additional, more broadly available input data (Kampffmeyer, Salberg, and Jenssen 2018; Malkin et al. 2019; Schmitt et al. 2019); however, all existing land cover models are biased by the geographic locations on which they were trained. Large systematic errors in predictions limit their applicability and are challenging to detect at scale.[1] Finally, the classification tasks are constantly shifting. While one dataset may segment vegetation simply into "low vegetation" and "tree canopy", other applications may require delineating coffee farms from orchards.

To a Geographic Information Systems (GIS) professional, however, land cover mapping is an inherently human-driven process augmented by technology. Accurate and useful labels themselves, not a training dataset for ML algorithms, are the immediate goal. The process typically starts with color-based segmentation algorithms that create initial maps, followed by experts who provide labels in different areas, creating rules on the fly, and then manually correcting the remaining errors. The labor efficiency of the process may increase as the humans learn how to use these tools better, but is not boosted by quick adaptation of the classification algorithms themselves.[2] This makes land cover mapping at the resolution and scale needed today cost-prohibitive for most agencies.

A hybrid system for accurate and efficient land cover labeling would more tightly integrate the human and machine efforts. Here we investigate a land cover mapping workflow where users' work immediately affects the performance of prediction algorithms.

Our design, which incorporates human feedback integrated in real time as training points for our model, can be seen as an instance of machine teaching (Simard et al. 2017; Zhu 2015), as humans deploy their own intelligence to identify and correct mislabeled points in an effort to improve the model. However, our system does not attempt to create an autonomous entity, capable of generalizing, as the final result: the ability to efficiently label large areas is the goal, and the final trained algorithm is but one aspect of the overall workflow. To a human, the ML model is simply a powerful macro that they (re)define on the fly in order to amplify their work. To the ML model, the human is the source of data to learn from. Together, this hybrid system holds the potential to outperform existing GIS workflows as well as pure ML approaches in cost and accuracy.

[1]For example, imagery of the contiguous US at 1m resolution covers 8 trillion pixels.

[2]Typically, separately tuned random forests are used, although neural networks are rapidly gaining traction.

3 Land cover study design

We focus on the following task: given a pretrained segmentation model, which was trained on 1m-resolution imagery and a four-class land cover map of Maryland (Chesapeake Conservancy 2017), we would like to quickly (within at most 15 minutes) produce accurate maps for regions of 1m-resolution imagery in New York State. This change in the geographic region where the model is to be applied represents a *domain shift*. We aim to create the map of each region by slightly changing the parameters of the Maryland model to fit a limited number of guidance points in the new areas.

We vary two parameters in our study: the **fine-tuning method** and **query method**.

The **fine-tuning method** is the algorithm for retraining the model to fit new guidance points. Such a method needs to be fast and sample-efficient. As we have ground truth data in the entire Chesapeake watershed, including Maryland and N.Y., various choices for fine-tuning can be evaluated offline.

The **query method** is the method for selecting guidance points on which to fine-tune the model on a new region. The main object of our study is to compare automatic methods, such as random selection or active learning approaches, to **hybrid (human-guided) methods**, where users iteratively view the current model's predictions, correct the labels at points of their choice, and trigger model retraining. The traditional active learning approaches to automatic selection of points to query can also be studied offline on a fully labeled dataset (Sec. 4).

We implement the **hybrid (human-guided) method** by developing a web tool that allows users to iterate between labeling and testing the model (Fig. 2). The tool exploits the spatial nature of the data in the task, allowing the user to zoom and pan in the high-resolution imagery of an area to find areas where they want to test the current algorithm. Upon a click on the map, the prediction of the current model on a surrounding 500m$\times$500m patch of land is overlaid on the map. The user can then label pixels of their choice, either where they see errors or for some other reason they think that the label will be useful. They can induce near-instant retraining of the model at any time with the click of a button. After that, they can check how well the retrained model works by clicking on the imagery again.

Base segmentation model

Our base segmentation model takes input patches of high-resolution (1m) four-band aerial imagery from the USDA National Agriculture Imagery Program (NAIP) and outputs a segmentation of the image (per-pixel classification) over four land cover classes (water, forest, field, impervious surfaces). The default training label datasets are from (Chesapeake Conservancy 2017).

The model is a modified U-Net model (Ronneberger, Fischer, and Brox 2015; Rakhlin et al. 2018) (a type of convolutional neural network) that contains four down-sampling and four up-sampling layers and skip connections between them. For down-sampling, we use a simple 2$\times$2 max-pooling. For up-sampling, we use deconvolution (transposed convolution). Before each down-sampling and up-sampling layer,

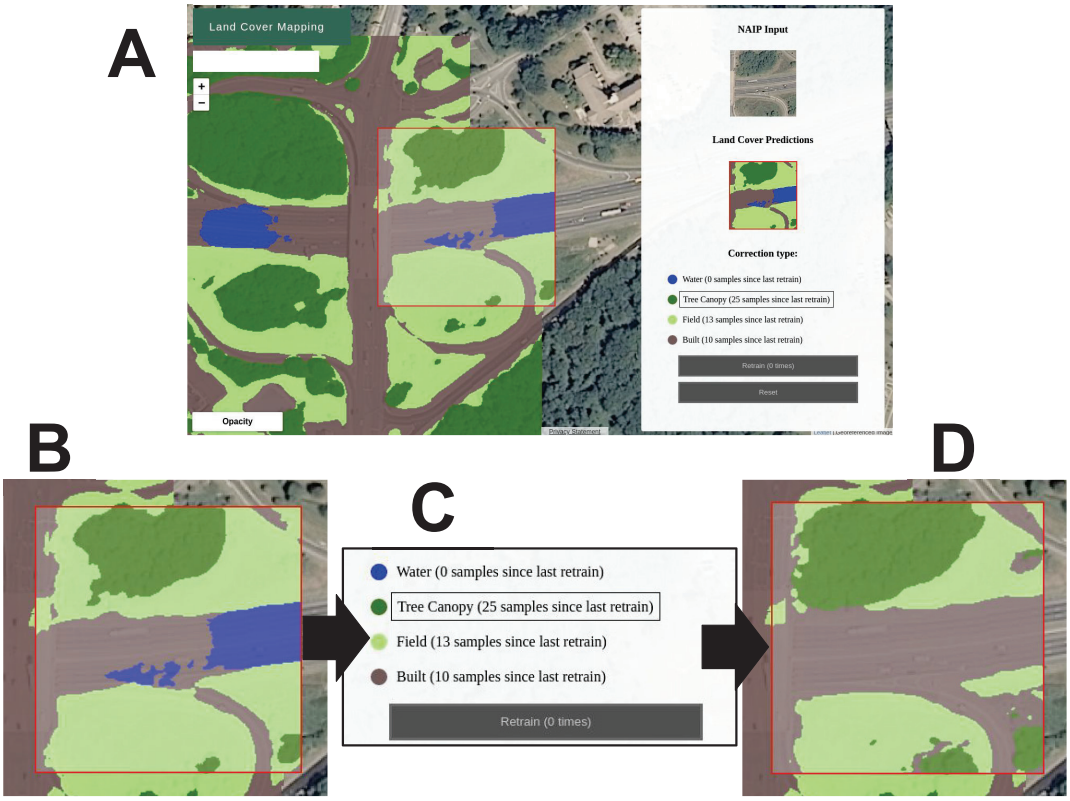

Figure 2: User interface of our land cover labeling tool. (**A**) Land cover prediction results are overlaid on top of the map. (**B**) The user can easily identify misclassified pixels and (**C**) submit corrections by clicking on the map. (**D**) Pressing "Retrain" updates the model and displays new land cover predictions in the interface. In this example, the user provided a handful of point corrections in the impervious surface initially misclassified as water.

we insert two convolutional layers. The first two convolutional layers have 32 3×3 filters. Group normalization (Wu and He 2018) is applied after the second convolution in every layer followed by ReLU. Valid padding is used in all layers making the predicted output smaller than the input. The number of filters is doubled after each pooling layer, the representational bottleneck layers use 512 3×3 filters. We trained the network for 100 epochs on $\sim$ 90000 randomly selected image patches of size 240 x 240 sampled from the state of Maryland. We used the Adam optimizer (Kingma and Ba 2014) with cross-entropy as segmentation loss and an initial learning rate of 0.001 decaying to 0.0001 after 60 epochs.

Formally, given parameters θ and an image $X = \{x_{ijk}\} \in \mathbb{R}^{w \times h \times c}$ (where $c = 4$ is the channel depth and $w \times h$ are the image dimensions), the model outputs a probability distribution over the target classes at each pixel, i.e., $f(\theta, X) \in \mathcal{D}(n)^{w \times h}$, where $\mathcal{D}(n)$ is the probability simplex on the $n = 4$ output classes. This yields distributions over labels $P_\theta(\hat{y}_{ij}|X)$ for each coordinate (i, j).

4 Offline active learning experiments

As discussed in Sec. 3, we investigate different methods for fine-tuning a pre-trained model and querying for new label data in a different domain. In these experiments, and in the online experiments described in Section 5, the new domain is imagery from four 84km^2 *areas* in New York. Our offline experiments are meant to identify the optimal fine-tuning and query methods, which are then used in online user studies. In our offline experiments, the base segmentation model is adapted to a small number – 10 to 2000 – of *automatically chosen* labeled pixels (less than 0.01% of each target area). Then the performance is evaluated on the entirety of the target areas.

Fine-tuning methods

The following **fine-tuning methods** were tested:

LAST k LAYERS Following (Yosinski et al. 2014), the final k convolutional layers in the U-net architecture have their weights exposed as trainable via gradient descent (initialized from the weights of the base model), while all other parameters in the network are held fixed. Here, $k \in \{1, 2, 3\}$.

GROUP NORMALIZATION PARAMETERS Inspired by the success of feature-wise transformations (Dumoulin et al. 2018) in neural style transfer (Dumoulin, Shlens, and Kudlur 2016) and visual question answering (Perez et al. 2018), we extended it for model fine-tuning. Our U-net

Query method	Last 1 Layer Acc	Last 1 Layer IoU	Last 2 Layers Acc	Last 2 Layers IoU	Last 3 Layers Acc	Last 3 Layers IoU	Group Params Acc	Group Params IoU	Dropout Acc	Dropout IoU
Baseline	0.725	0.510	0.725	0.510	0.725	0.510	0.725	0.510	0.725	0.510
Random	0.806	0.608	0.825	0.677	0.824	0.658	0.791	0.562	0.787	0.597
Entropy	0.736	0.501	0.731	0.587	0.765	0.572	0.760	0.520	0.741	0.550
Min-Margin	0.811	0.608	0.834	0.701	0.832	0.685	0.793	0.580	0.785	0.601
Mistakes	0.729	0.551	0.781	0.631	0.756	0.621	0.787	0.575	0.762	0.609

Table 1: Results of fine-tuning on 400 points selected by different query methods, averaged over four target areas and five random seeds.

architecture uses group normalization (Wu and He 2018) in the final convolutional layers. The group normalization parameters affect large groups of filters in each layer via a single affine transformation, with the assumption that filters within a group are correlated. Thus, training these parameters to fit new training points causes correlated changes in the layers' outputs, providing a regularized mechanism to affect the entire network, in contrast with full backpropagation, which affects all weights in the chosen layers.

DROPOUT We effect dropout, i.e., set the outputs of a fixed subset of the neurons to 0, in the final k convolutional layers. Searching for the binary mask that minimizes a loss is a discrete optimization problem, which we solve using a simple genetic algorithm. Here we use $k = 5$ and a mean dropout rate of 0.2, but we conducted only limited experiments due to the high cost of this method, which requires evaluation of the model at all sample points at each of 64 mutation iterations.

In our experiments the **Last k Layers** and **Group Params** methods are implemented using the Adam optimizer for 10 epochs, $\epsilon = 10^{-5}$. Learning rates were set as follows: 0.01 for last 1 layer, 0.005 for last 2 layers, 0.001 for last 3 layers, and 0.0025 for group parameters.

Query Methods

Motivated by (Zhang 2017; Settles 2012), we also investigated three **query methods** for selecting the additional 10 to 2000 labeled pixels used by the fine-tuning methods:

RANDOM Sample points (i^*, j^*) uniformly randomly from the training area.

ENTROPY Select points which maximize the Shannon entropy of output distributions over classes:

$$(i^*, j^*) = \operatorname{argmax}_{(i,j)} \left(-\sum_{\ell} P_\theta(\hat{y}_{ij} = \ell | X) \log P_\theta(\hat{y}_{ij} = \ell | X) \right).$$

MIN-MARGIN Select points which minimize the difference between probabilities assigned to the most-likely and second-most-likely classes:

$$(i^*, j^*) = \operatorname{argmin}_{(i,j)} \left(P_\theta(\hat{y}_{ij} = \ell^1_{ij} | X) - P_\theta(\hat{y}_{ij} = \ell^2_{ij} | X) \right),$$

where ℓ^1_{ij} and ℓ^2_{ij} are the two most likely classes under $P_\theta(\hat{y}_{ij} | X)$.

We also include the following method, the purpose of which is to make a comparison with *humans* selecting mistake points in our online study. It is *not* an automatic query strategy, as it assumes the model has access to an all-knowing labeling oracle *before* it chooses where to query the oracle for labels. It simply imitates a teacher that feeds randomly chosen mistake points to the model.

MISTAKES Uniformly sample points (i^*, j^*) where the model's prediction disagrees with the ground truth.

Results

Because it is prohibitively costly to select points using the ENTROPY, MIN-MARGIN, and MISTAKES methods at *every* training iteration, we approximate this procedure by batching: periodically evaluating the model on the training area and selecting the optimal points among a large set of 10000 uniformly sampled locations. Namely, we evaluate the model and select a new batch of points after 10, 40, 100, 200, 400, 1000, and 2000 points have been chosen.

The experiments are repeated five times with different random seeds for each combination of the adaptation method, point selection strategy, and target area. The average adaptation performance when methods use only 400 labeled pixels – close to the number labeled by users in our online studies – is shown in Table 1, while the variation in accuracy across the whole range of additional training points is shown in Figure 3. (See also the supplemental materials - https://aka.ms/human-machine-2020-si - for the full set of curves.)

For all fine-tuning methods, we observed a similar ranking of the performance of active learning query methods, with MIN-MARGIN performing best, but only slightly better than RANDOM, and ENTROPY performing worst. Most interestingly, the MISTAKES method performs significantly worse than RANDOM: even giving the model access to ground truth knowledge does not improve performance. On the other hand, in online experiments (Sec. 5), we show that replacing this mock "uniform teacher" with a human teacher *does* improve performance.

5 Online study of the hybrid labeling system

As can be seen from the indicated confidence intervals in Fig. 3, it is not clear that we can expect any of the active

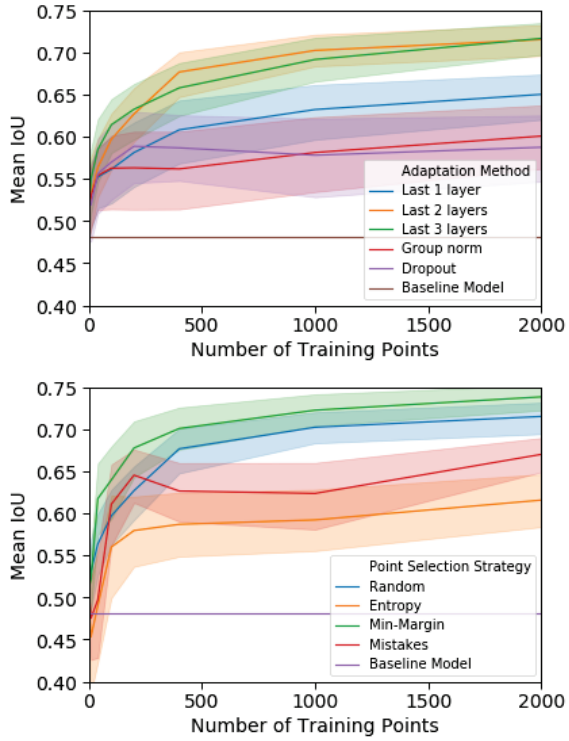

Figure 3: Performance of different **fine-tuning** methods (top) and **query methods** (bottom), mean and standard deviation over 5 runs and 4 target areas. At several stages – after 10, 40, 100, 200, 400, 1000, and 2000 points have been seen – the system selects a further set of training points using the given query method and retrains the model using the fine-tuning method. The performance of the model evaluated on the entire target region tends to improve as more points are seen.

learning methods to outperform random selection of data points to label, as was previously often observed in active learning literature (Settles 2011). Thus, we test our hybrid labeling system – the HUMAN query method – against the RANDOM point query method. As the LAST 1 LAYER and LAST 2 LAYERS fine-tuning methods tend to perform best in the offline experiments, we also choose them for use in online experiments. [3]

Setup

We recruited 50 users[4] through Amazon Mechanical Turk to implement the HUMAN method using the web interface and interactions described in Sec. 3. Users use the web tool in a series of 15-minute *tasks*. A *task* is performed in one of four distinct 84km^2 *areas* in New York and using one of two fine-tuning methods chosen above. Before each *task*, the model is reset to the baseline, pretrained only on data from Maryland. Each user performs four *tasks* (one for each *area*, in a random order): in the first three *tasks*, the user uses one type of fine-tuning method, while in the fourth *task* the other fine-tuning method is used. Such an assignment allows us to separate the first task – during which the user is getting used to the tool – from tasks 2 and 3, where the user is assumed to be doing their best work, and from task 4, where the learning system changes its behavior (i.e. where the *fine-tuning method* changes). This allows us not only to measure the variation in performance across users, *fine-tuning method*, and *areas*, but also to see if the users are building an understanding of how the model and its adaptation work. Users are able to query the model, submit labels, and retrain the model at will during each task (see Sec. 3).

We use a standard crowdsourcing setup in the Amazon Mechanical Turk system to acquire unbiased ground truth labels on the same four *areas* in New York – we refer to these labels as the "crowdsourced ground truth labels"[5]. We collected a total of 6009 labels on randomly selected points, from 54 unique labelers, resulting in a dataset of 3441 unambiguous labeled points. These labels agree with the Chesapeake ground truth data 91.1% of the time, which is in line with that data product's published quality estimates (Chesapeake Conservancy 2016).

Now, during each *task*, every time the user induces retraining of the model, we calculate that model's performance on the set of crowdsourced ground truth labels from the area in which they are working. We compare this method with the RANDOM query method using the crowdsourced ground truth dataset. In the crowdsourced labeling task, users take $\sim$ 3 seconds to label each pixel they are shown. Thus, in a 15-minute window, they could provide labels on $\sim$ 300 randomly sampled points. A central question is that of *label efficiency*: is human time and money best spent by labeling the central pixels of random patches of aerial imagery (human as *label oracle*) or by using our interactive tool (human as *query method* and *label oracle*)?

Results

The subplots in Figure 4 show accuracy and mean intersection-over-union (IoU) of intermediate models achieved at different times in the 15 minute fine-tuning sessions, averaged across users. In the case of the RANDOM method, we assume that 300 points are added at uniform time intervals and the model is retrained every 45 seconds. As the model for a specific user will fluctuate in performance over the duration of a single session – users pick up on different deficiencies in the base model at different points during a session – we summarize the HUMAN method as a whole by averaging performance metrics over sets of users. Models fine-tuned using the HUMAN query method consistently outperform models that are fine-tuned with RANDOM queried points, within 3 minutes of labeling ($\sim$ 60 samples).

[3] Precisely, LAST 1 LAYER was full adaptation of the 64 $\times$ 4 parameters in the last (softmax) layer (gradient descent to convergence on all user-supplied points), while LAST 2 LAYERS was a fixed number of iterations of gradient descent on the parameters of the last two layers.

[4] See the supplemental materials - https://aka.ms/human-machine-2020-si - for more details on the study setup.

[5] See the supplemental material for further details about the "crowdsourced ground truth labels.

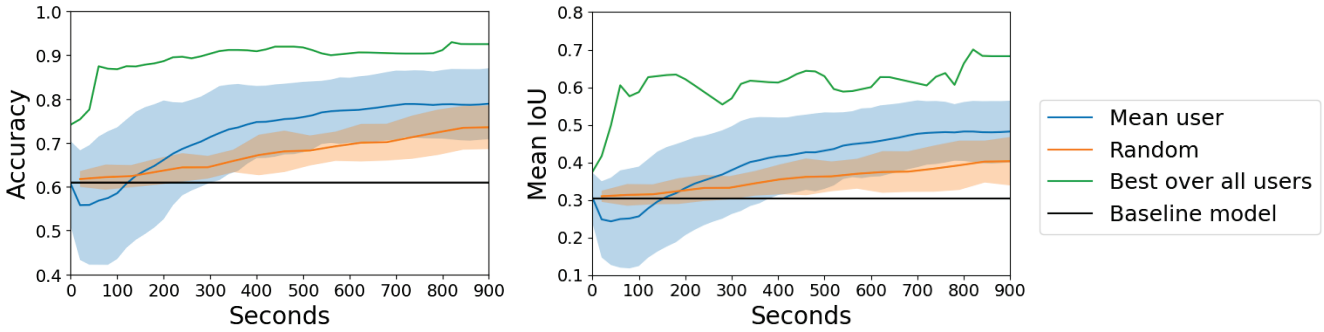

Figure 4: Performance of HUMAN and RANDOM query methods for model fine-tuning in a 15-minute time window, measured in pixel accuracy (**Left**) and mean IoU (**Right**). Mean user performance is calculated over the top 50% of users and considers sessions using the LAST 2 LAYER fine-tuning method. Random performance is averaged over 10 seeds, with points assumed to be added every 3 seconds. Both methods are averaged over the same four *target areas*.

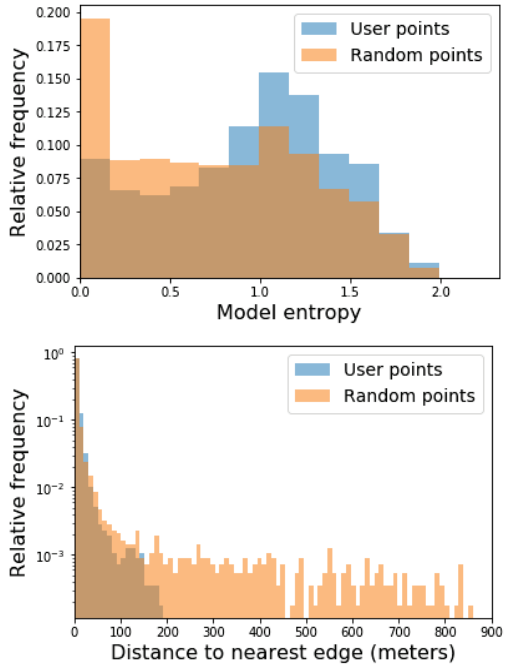

Figure 5: Distribution of HUMAN points versus RANDOM points in terms of the base segmentation model's class entropy at each point (**Top**) and the distance to nearest (Canny) edge at each point (**Bottom**).

The top curve in Fig. 4 shows the best model over all users at each point in time, showing that some expert users are in fact able to dramatically outperform RANDOM.

Further analyzing our user results across the four consecutive tasks, we find that users' area-adjusted performance in task 2 is highly predictive of their performance in task 3 ($p < 0.01$, rank-correlation $\rho = 0.4$): of the top 25 (half) of users ranked by (IoU) performance in task 2, 17 are also among the top 25 in task 3. Thus, **the better-performing labelers are detectable** in a statistically significant manner. This indicates that the users are developing different levels of intuition about the inner workings of the network and the fine-tuning method. In addition, the performance of the users in tasks 2 and 3 is far less predictive ($\rho = 0.1$) of their performance in task 4, where the fine-tuning method is switched.

An analysis of points submitted by all users in an area show that the users are *not* choosing points to label at random. Different users are drawn to similar parts of the study areas, while other parts remain unlabeled by most users. Figure 5 shows the distribution of points added by users compared to those sampled randomly in terms of their distance to the nearest canny edge in the underlying imagery, and the class entropy predicted by the base segmentation model at the points. We observe that these distributions are distinctively different; users select points that are closer to edges and are more likely to select points from the mid-entropy ranges. We visualize the distribution of user points in the supplemental material - https://aka.ms/human-machine-2020-si.

Our offline experiments with the MISTAKE method indicate that the model simply knowing where its errors are cannot automatically beat the RANDOM selection of points for labeling. This indicates that **human guidance goes beyond simply quickly spotting errors**, especially for best performers, reminiscent of the super-teacher idea (Ma et al. 2018). Text feedback from users (see supplemental material) provides further interesting insights that should be useful in the design of hybrid systems of this kind.

6 Discussion

We have conducted a study of hybrid human-AI intelligence on the task of high-resolution land cover mapping. We demonstrate that giving control of the data selection process to the human yields significant improvements in model accuracy - compared to strictly algorithmic methods - in the land cover mapping task.

Our results show that, by injecting the human into the learning loop, gains from both the human and the AI labor are amplified, not replaced. For the machine, sparse but well-

chosen human feedback reduces the cost of computational resources needed to adapt models. For the human, increased sample efficiency of the ML systems acts like an ever-more useful wand with which they can paint land cover. Together, this collaboration achieves critical cost reduction in practical problems. The Chesapeake Bay dataset was created in 10 months at a cost of $1.3 million, though it covers just 2% of the US (Chesapeake Conservancy 2017) with an estimated accuracy of 90%-95%. The best user from our study, averaged over the four target areas, achieved an accuracy of 89.1% in just one hour of labeling work.[6] If such users were to label the entire Chesapeake Bay watershed using our method, this would take 925 hours of work at a labor cost of $18.5k. Of course, other tradeoffs between accuracy and cost are possible by allowing users to work longer on each area or even to work collaboratively.

We hypothesize that the performance improvements we observe in the HUMAN method are due to users developing a *theory of mind* for the ML system - learning to understand the workings of a particular AI algorithm (in the context of a given task) and therefore learning which labels it would benefit from observing. Our online experiment tests for this property in a single dimension - we measure user performance across 4 tasks, however in the 4$^{\text{th}}$ task we switch the fine-tuning method without informing the users. We do observe a decrease in performance in the 4$^{\text{th}}$ task, however cannot conclude that this drop in performance is because users have formed a theory of mind for the first fine-tuning method, that is subsequently broken by switching the task. Follow up studies should test whether this property holds when controlling for more variables in the same setting (e.g. time taken for fine-tuning or different measures of task engagement), and whether it holds in other settings (e.g. in tasks of different complexities and running times). Crowdsourced workers have been shown to be effective even in complicated tasks and have been utilized in combination with ML models to achieve large-scale labeling goals (Kamar, Hacker, and Horvitz 2012). Understanding *how* humans interact with, and indeed, can collaborate with ML models is thus an important component of tackling other large scale problems.

Complementary to the *why* questions surrounding human-machine collaboration are practical questions of *what works best* in different settings. Our proposed framework requires both a **fine-tuning method** and **query method** and future work should explore the interaction between choices made to implement these parts. Our offline experiments aimed to test a diverse set of **fine-tuning methods**: LAST k LAYERS expose a relatively large set of parameters to be fit with SGD based methods, GROUP NORMALIZATION PARAMETERS, in contrast, exposes an (engineered) set of few parameters that are also fit with SGD based methods, while DROPOUT uses a local-search algorithm over discrete choices (dropout patterns) to fit newly labeled data. Other choices that can be tested in diverse application settings include: fitting smaller non-linear models (e.g. random forests) using the base segmentation model as a feature extractor, or deciding on sets of parameters (throughout the entire model) to fit in an offline study. Design decisions for the interface that human labelers use should also be considered thoroughly. For example, our interface requires the human labelers to manually trigger a model retraining, however this could be performed automatically with some set frequency. Increasing this frequency tightens the feedback loop that users will experience between submitting labels and observing the effects on model performance.

In all cases, in problems of massive scale where unlabeled data is practically limitless, such as land cover labeling, it is not likely that a few months of labeling through our tool would create enough training data that the need for human labor would disappear. Instead, applications that are now infeasible, such as quick generalization to new areas or addition of new target classes (shown in supplemental materials), would become feasible, making both the ML algorithms and human labor more valuable than before.

[6]This number is in line with the recent state-of-the-art algorithm (Malkin et al. 2019) which uses 30m low-resolution labels as additional data. Our approach does not rely on the existence of such low-resolution labels.

Supplemental Materials

The supplemental material for this paper can be found at https://aka.ms/human-machine-2020-si.

Acknowledgements

The authors thank Lucas Joppa and the Microsoft AI for Earth initiative for their support and all reviewers for their helpful comments. C.R. was partially supported by the NSF grant CCF-1522054 (COMPUSTNET: Expanding Horizons of Computational Sustainability). B.D. was partially supported by NSF grants #1935451, #1914522, and #1763108.

References

Bachman, P.; Sordoni, A.; and Trischler, A. 2017. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*.

Cai, C.; Reif, E.; Hegde, N.; Hipp, J.; Kim, B.; Smilkov, D.; Wattenberg, M.; Viegas, F.; Corrado, G.; Stumpe, M.; and Terry, M. 2019. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *CHI Conference on Human Factors in Computing Systems*. ACM.

Castelluccio, M.; Poggi, G.; Sansone, C.; and Verdoliva, L. 2015. Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*.

Chesapeake Conservancy. 2016. High resolution lulc classification accuracy assessment methodology. https://www.chesapeakebay.net/channel_files/24793/lulcaccuracyassessment_detailed_methodology.pdf.

Chesapeake Conservancy. 2017. Land cover data project. https://chesapeakeconservancy.org/wp-content/uploads/2017/01/LandCover101Guide.pdf.

Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; and Raskar, R. 2018. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Dumoulin, V.; Perez, E.; Schucher, N.; Strub, F.; Vries, H. d.; Courville, A.; and Bengio, Y. 2018. Feature-wise transformations. *Distill*. https://distill.pub/2018/feature-wise-transformations.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

Facebook. 2019. Mapping the world to help aid workers with weakly semi-supervised learning. https://ai.facebook.com/blog/mapping-the-world-to-help-aid-workers-with-weakly-semi-supervised-learning/.

Griffiths, T.; Abbott, J.; and Hsu, A. 2016. Exploring human cognition using large image databases. *Topics in Cognitive Science* 8(3):569–588.

Hansen, M. C.; Potapov, P. V.; Moore, R.; Hancher, M.; Turubanova, S.; Tyukavina, A.; Thau, D.; Stehman, S.; Goetz, S.; Loveland, T.; et al. 2013. High-resolution global maps of 21st-century forest cover change. *Science* 342(6160):850–853.

Hsu, W.-N., and Lin, H.-T. 2015. Active learning by learning. In *Twenty-Ninth AAAI conference on artificial intelligence*.

Huang, T.-K.; Li, L.; Vartanian, A.; Amershi, S.; and Zhu, J. 2016. Active learning with oracle epiphany. In *Advances in Neural Information Processing Systems*, 2820–2828.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 467–474. International Foundation for Autonomous Agents and Multiagent Systems.

Kampffmeyer, M.; Salberg, A.-B.; and Jenssen, R. 2018. Urban land cover classification with missing data modalities using deep convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11(6):1758–1768.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kuo, T.-S.; Tseng, K.-S.; Yan, J.-W.; Liu, Y.-C.; and Wang, Y.-C. F. 2018. Deep aggregation net for land cover classification. In *Computer Vision and Pattern Recognition (CVPR) workshops*.

Ma, Y.; Nowak, R.; Rigollet, P.; Zhang, X.; and Zhu, X. 2018. Teacher improves learning by selecting a training subset. In *International Conference on Artificial Intelligence and Statistics*, 1366–1375.

Malkin, K.; Robinson, C.; Hou, L.; Soobitsky, R.; Czawlytko, J.; Samaras, D.; Saltz, J.; Joppa, L.; and Jojic, N. 2019. Label super-resolution networks. In *International Conference on Learning Representations (ICLR)*.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; Gorur, D.; and Lakshminarayanan, B. 2019. Do deep generative models know what they don't know? In *International Conference on Learning Representations*.

Nushi, B.; Kamar, E.; and Horvitz, E. 2018. Towards accountable ai: Hybrid human-machine analyses for characterizing system failure. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. AAAI.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Rakhlin, A.; Neuromation, O.; Davydow, A.; and Nikolenko, S. 2018. Land cover classification from satellite imagery with u-net and lovász-softmax loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 262–266.

Robinson, C.; Hou, L.; Malkin, K.; Soobitsky, R.; Czawlytko, J.; Dilkina, B.; and Jojic, N. 2019. Large scale high-resolution land cover mapping with multi-resolution data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Schmitt, M.; Hughes, L. H.; Qiu, C.; and Zhu, X. X. 2019. Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. *arXiv preprint arXiv:1906.07789*.

Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.-F.; and Dennison, D. 2015. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*.

Settles, B. 2011. From theories to queries: Active learning in practice. In Guyon, I.; Cawley, G.; Dror, G.; Lemaire, V.; and Statnikov, A., eds., *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, 1–18. Sardinia, Italy: PMLR.

Settles, B. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(1):1–114.

Simard, P. Y.; Amershi, S.; Chickering, D. M.; Pelton, A. E.; Ghorashi, S.; Meek, C.; Ramos, G.; Suh, J.; Verwey, J.; Wang, M.; et al. 2017. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742*.

Tian, C.; Li, C.; and Shi, J. 2018. Dense fusion classmate network for land cover classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 192–196.

Wu, Y., and He, K. 2018. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Yang, Y., and Newsam, S. 2010. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 270–279. ACM.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, 3320–3328.

Zhang, H.; Qi, Z.-f.; Ye, X.-y.; Cai, Y.-b.; Ma, W.-c.; and Chen, M.-n. 2013. Analysis of land use/land cover change, population shift, and their effects on spatiotemporal patterns of urban heat islands in metropolitan shanghai, china. *Applied Geography* 44:121–133.

Zhang, C.; Tavanapong, W.; Kijkul, G.; Wong, J.; de Groen, P. C.; and Oh, J. 2018. Similarity-based active learning for image classification under class imbalance. In *2018 IEEE International Conference on Data Mining (ICDM)*, 1422–1427. IEEE.

Zhang, C. 2017. *Active Learning and Confidence-rated Prediction*. Ph.D. Dissertation, UC San Diego.

Zhu, X. 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.