Proceedings of the Ninth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2021)

Iterative Human-in-the-Loop Discovery of Unknown Unknowns in Image Datasets

Lei Han [1], **Xiao Dong** [2], **Gianluca Demartini** [1]

[1] The University of Queensland, Brisbane, Australia
[2] Sun Yat-sen University, Guangzhou, China

Abstract

Automatic predictions (e.g., recognizing objects in images) may result in systematic errors if certain classes are not well represented by training instances (these errors are called unknowns). When a model assigns high confidence scores to these wrong predictions (this type of error is called unknown unknowns), it becomes challenging to automatically identify them. In this paper, we present the first work on leveraging human intelligence to discover unknown unknowns (UUs) in an iterative way. The proposed methodology first differentiates the feature space generated by crowd workers labelling instances (e.g., images) in an active learning fashion from the space learned by the prediction model over a batch training phase, and thus identifies the predictions most likely to be UUs. Next, we add crowd labels collected for these discovered UUs to the training set and re-train the model with this extended dataset. This process is then repeated iteratively to discover more instances of both unknown and underrepresented classes. Our experimental results show that the proposed methodology is able to (i) efficiently discover UUs, (ii) significantly improve the quality of model predictions, and (iii) to push UUs into known unknowns (i.e., the model makes mistakes but at least its classification confidence on those instances is low so those predictions can be discarded or post-processed) for further investigation. We additionally discuss the trade-off between prediction quality improvements and the human effort required to achieve those improvements. Our results bear implications on building cost-effective systems to discover UUs with humans in the loop.

1 Introduction

Supervised classification usually works under a close-world assumption (Reiter 1981), where the predicted classes are estimated based on training data. Attenberg, Ipeirotis, and Provost (2015) argued that for classes that are not covered by training data, the models may result in systematic errors of mis-classification. These errors are called *unknowns* as the trained model cannot make effective predictions for unseen classes. Lewis and Catlett (1994) proposed an uncertainty sampling strategy for supervised learning, which selects instances with uncertain labels and presents them to human experts to label. Such strategy, however, does not work when a model assigns *high confidence scores* to wrong predictions as these classification decisions are not uncertain and thus are not able to be selected for further investigation. This type of error, which is very difficult to detect due to the high confidence of the model, is called *unknown unknowns* (UUs) since the classification model is not aware that such mistakes have been made. Figure 1 shows an example of UUs that cannot be identified by the classification model. In this example, a model is trained on a set of images for three classes: zebra, bird and train. Thus, the model wrongly predicts an image of airplane being of class "bird" as the class "airplane" is not covered by the training data. The confidence score made by the model, however, is high (i.e., 0.999 for this prediction). Therefore, it becomes challenging to automatically recognize such incorrect classification decisions (also known as UUs).

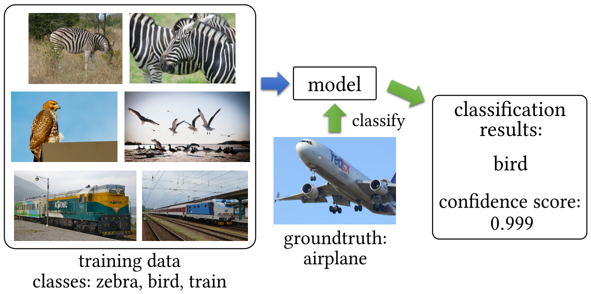


Figure 1: An example of UUs where the model makes a *wrong* classification but with a high confidence score. In this case, the classification model is not able to identify such mistakes automatically.

Recently, evaluation initiatives have been run for the UU identification task[1]. Prior approaches proposed to address this problem can generally be grouped into two classes: algorithmic and crowdsourced. In algorithmic UU detection approaches, the selection of potential UU candidates is based on the assumption that systematic biases would lead to clusters of blind spots in the feature space. Lakkaraju et al. (2017) proposed a partitioning scheme to cluster instances in the test set and a multi-armed bandit algorithm to pick up UUs that maximize utility functions. This work

[1] https://cats4ml.humancomputation.com/overview

has later been extended by Bansal and Weld (2018) to assign a higher utility value to the selected UUs that give a better coverage of the test dataset, which they call a greedy algorithm to discover UUs. Crowdsourced solutions, on the other hand, leverage human intelligence to add data samples that do not exist in the training data. For example, the "beat-the-machine" model (Attenberg, Ipeirotis, and Provost 2015) takes an open-world assumption and encourages human participants to submit instances that are missing in training data by rewarding contributions proportionally to the magnitude of system failures. Vandenhof (2019) proposed a "contradict-the-machine" model that asks crowd workers to generate examples that follow the model decision rules but that lead to a wrong classification result. Liu et al. (2020) proposed an approach where humans define patterns that are used to identify candidate UUs.

These existing solutions, however, have limitations. First, algorithmic selection of potential (partitions of) UUs only covers a subset of the test dataset (with top-*n* utility values) while in the reality there may be a more diverse set of UUs that are distributed sparsely in none of the selected partitions and thus would be skipped. Therefore, we argue that the discovery of UUs should investigate all the instances for which predictions are made with high confidence scores (given that the definition of UUs refers to high confidence classification errors). Second, asking human workers to manually generate classification failure examples is not realistic for non-textual data (e.g., images), and even looking for this material in the "open world" (e.g., by means of web search) is challenging and makes the scalability of such solutions very limited. In fact, to understand the content of non-textual data (e.g., image annotation and captioning), a number of approaches have been proposed to automatically extract semantics from multimedia content with or without humans-in-the-loop (Jing et al. 2016; Venugopalan et al. 2017; Andriluka, Uijlings, and Ferrari 2018; Wu et al. 2018). As the way in which humans understand multimedia content (e.g., recognizing objects in images) may be influenced by the context in which the content is presented (Zhang et al. 2019), introducing a human input component in the system allows us to build more human-like algorithms. Following this technique, we also propose to leverage human effort to discover UUs. To this end, by involving humans in the loop, instead of asking crowd workers to generate new instances, we rather leverage human intelligence to select instances from the existing dataset and extend the training data with new labels which are then used to re-train the model. Considering the high cost and potentially low quality of crowd work, the key challenges of such an approach lie in deciding (i) how much human work is enough to reliably discover UUs, and (ii) how to deal with noisy crowdsourced data for this particular task. Since UUs are caused by under-representation or absence of certain classes in the training set, we believe that providing better coverage for these classes would effectively address the problem.

Inspired by recent work (Yang et al. 2019), which employed humans to inspect potentially wrong labels in training datasets, we take a human-in-the-loop approach to identify what is missing or is under-represented in the training data, and solve the UU detection problem in an *iterative* way. Specifically, at each iteration, we differentiate the feature spaces generated by crowd workers labelling new instances in an active learning fashion from the space learned by the predictive model over the available training data, and thus identify the predictions most likely to be UUs. Next, we add these crowd generated labels for the discovered UUs into the training set and re-train the model with this extended dataset. This process is repeated iteratively to discover more instances of both unknown and under-represented classes. Our experimental results show that the proposed method is able to efficiently discover UUs and significantly improve the quality of model predictions. We also show that our method can push UUs into *known unknowns*, where the model still makes mistakes but, at least, its confidence score for those instances is low so those predictions can be discarded or post-processed.

To summarize, this work makes the following three-fold novel contributions:

- We present an *iterative* approach in an active learning manner for UU detection. By differentiating the latent space generated by human input from the feature space learned by the predictive model, we are able to understand the difference between human and machine classifications and to identify the problematic predictions which are likely to result in UUs. We then add labels for these discovered UUs to the training set and re-train the model. This process is repeated to discover more UUs.
- We look at the *cost-efficiency trade-off* between prediction quality improvement and human effort required to achieve that improvement. Due to the high cost of employing human labor, extra manual effort may not be beneficial if it no longer contributes to the improvement of the model prediction quality. This contributes to building cost-effective workflows to discover UUs with humans in the loop.
- We show that our approach not only is able to improve the quality of the predictions by efficiently discovering UUs, but it can also effectively *lower the model confidence* on problematic predictions, pushing potential UUs into uncertainty (e.g., known unknowns based on the reduced confidence score of UU instances). Thus, these instances can be automatically selected for post-processing to improve model classification quality.

To the best of our knowledge these three contributions have not been made by any previous related research looking at human-in-the-loop UU detection. The rest of this paper is structured as follows. Section 2 provides an overview of previous research on the discovery of UUs and on leveraging human effort to improve machine learning performance in dealing with both textual and multimedia content. In Section 3, we present our methodology that iteratively discovers UUs with humans-in-the-loop. In Section 4 we report experimental evaluation results to understand the effectiveness and budget-efficiency of our method. In Section 5, we discuss our key observations and their implications on building cost-effective systems to improve the performance of algorithmic classification. Finally, in Section 6 we summarize our findings and draw conclusions.

2 Related Work

This section presents previous research related to UU discovery and leveraging human intelligence to improve machine learning practice.

2.1 Discovering Unknown Unknowns

Discovering UUs has recently attracted research attention because of the traditional close-world assumption (Reiter 1981) taken by machine learning where a pre-defined model space and all the training data is required to be defined in advance. Andriluka, Uijlings, and Ferrari (2018) proposed to allow human annotators to decide the order in which elements in the image should be labeled. By doing this, the method allows them to address algorithmic errors first and to identify model unknowns. Attenberg, Ipeirotis, and Provost (2015) argued that for the cases not covered by the model space, there may exist systematic misclassification problems. Hence, they proposed the beat-the-machine (BTM) model leveraging crowd intelligence to find instances (with an "open-world" assumption) which have been missed while training the model. Vandenhof (2019) presented a hybrid model combining the advantages of both crowdsourcing and algorithms to identify UUs by uncovering model behaviors to crowd workers, and then asking them to use given templates to generate examples that the model would likely mis-classify. Despite the efficiency of these methods which is obtained by adding additional samples to the existing dataset, their scalability is limited as generating a new piece of non-textual materials (e.g., image or video) is challenging even in a crowdsourcing setup. Differently from existing work, we propose to leverage human effort to identify UUs in existing collections, and to add new labels for them into the training set. This is aimed at improving under-represented classes and shift towards a more balanced training dataset.

Complementing human-centered approaches, algorithmic solutions to identifying UUs have also been proposed. Lakkaraju et al. (2017) identified systematic classification bias and grouped test data into partitions on which they proposed a multi-armed bandit algorithm (UUB) to identify instances that lie in blind regions of the model space. Bansal and Weld (2018) argued that clustering techniques may result in an incomplete discovery of UUs, and thus introduced the concept of "effective coverage" to diversify utility values in identifying UUs that exist in other areas of the model space. Ramakrishnan et al. (2018) assumed UUs occur following observable features rather than at random, and proposed a framework based on reinforcement learning and human feedback to predict which instances are more likely to be UUs. Liu et al. (2020) extended the BTM model (Attenberg, Ipeirotis, and Provost 2015) and proposed a hybrid human-machine approach to identify UUs. Their method, known as patterned beat-the-machine (P-BTM), first asks human subjects to find patterns of UU instances and uses the discovered UU patterns to train a second classifier. This is then applied to identifying more UUs. As compared to them, the iterative approach we propose in our paper is more generalizable and applicable to cases in which defining patterns would be more challenging for humans like in the case of complex image datasets like the one we use in our experimental evaluation.

While these state-of-the-art methods have shown the possibility to automatically discover UUs, none of them performs the discovering process in an iterative manner like we do in this work, thus making the discovery incremental and cost-efficient. By investigating how the prediction quality can be improved and how the model confidence evolves through iterations as we add identified UUs into the training set and retrain the prediction model, we are able to understand the cost-effectiveness trade-offs in involving humans to discover UUs. Moreover, unlike existing methods, our approach is based on comparing two different feature spaces: (i) the space learned by the deployed classification model and (ii) the space generated by crowd worker contributions. We measure the similarity between the two spaces and are thus able to identify problematic classification decisions that are most likely to be UUs.

2.2 Human Effort in Machine Learning

In machine learning research, human input has often been leveraged to improve the quality of algorithmic output. For example, explanatory debugging (Kulesza et al. 2010) involves humans in the loop to help explain machine-learned program failures. Similarly, Nushi et al. (2017; 2018) involved humans to provide explanations for each failure and potential fixes within image captioning systems. Sroka and Braida (2005) and Zhang et al. (2019) have shown differences among humans and algorithms in understanding multimedia content such as speeches and images. They claimed that understanding how humans select features would be beneficial for building more human-like systems.

Previous work has also aimed at building iterative human-machine platforms to integrate human input into the learning processes. In computer vision, for example, iterative collaboration systems such as Flock (Cheng and Bernstein 2015) and Fluid Annotation (Andriluka, Uijlings, and Ferrari 2018) have been designed for capturing informative features from human input to generate hybrid machine learning classifiers or to understand the errors made by algorithms. On the other hand, human input is also leveraged in an active learning fashion (Joshi, Porikli, and Papanikolopoulos 2009; Siddiquie and Gupta 2010; Kovashka, Vijayanarasimhan, and Grauman 2011), where the algorithms first make initial predictions that are then corrected by humans. This process is repeated until certain criteria (e.g., model performance) are satisfied. For example, Rother, Kolmogorov, and Blake (2004) developed an iterative algorithm that uses human editing to correct initial the segmentation between image foreground and background. Yao et al. (2012) proposed an incremental learning approach that iteratively refines the model to detect presented objects in images and videos. Following this type of research, we also leverage human input to address the problem of UU detection in an iterative way. In addition, by measuring the improvement of the model quality, we are able to understand whether or not the cost of involving extra human work is justified by the improvement of model performance.

3 Methodology

To address the most challenging scenario where it is difficult for humans to generate examples for non-textual content (Attenberg, Ipeirotis, and Provost 2015; Vandenhof 2019), in our work we focus on the discovery of UUs in image datasets. Specifically, we define the task as identifying objects that cannot be recognized by the machine learning model but are actually present in the images. To this end, we use the MS COCO dataset (Lin et al. 2014) for our experiments as these images are complex and close to real world scenarios. For simplicity, we use images that are labeled as containing only one type of object[2]. We use images containing the six most frequently appearing objects in the dataset (i.e., giraffes, airplanes, zebras, clocks, birds, and trains), and define each type of object as a *class* that the model should identify in the images. This constitutes our entire image collection D. To create an initial training set (denoted by D_0) with unknown bias, we only use three classes from the original D to set up D_0 (i.e., the other three classes are manually removed to obtain D_0) while keeping all the six classes equally represented in the test set throughout the entire process[3].

Algorithm 1 presents the entire process of our iterative approach to discover UUs. In each iteration (Line 2 to 19), we first train a classifier to predict the object in the image, and thus obtain a set of features $\boldsymbol{f}_{mac}$ for each image and a confidence score c for each prediction p (Line 4). Then, we ask human subjects to describe the important objects represented in these test images asking them to write natural language text of a certain length (Line 5). Thus, we are able to extract features $\boldsymbol{f}_{hum}$ from the human generated text (Line 6), allowing us to compare human-generated features with the features learned by the model from training images (i.e., $\boldsymbol{f}_{mac}$). To differentiate the two feature spaces (Line 8 to 16), we pick up each image (denoted by m) as a "query" and simulate the task of searching for the most similar images. This allows us to understand how the two spaces differ from one another by looking at similar images. For a given image m, we assume that m has a higher chance to be a UU when the querying results are less similar by using $\boldsymbol{f}_{mac}$ as compared to using $\boldsymbol{f}_{hum}$ (detailed explanation in Section 3.2). Note that we skip images for which the prediction confidence is below a threshold τ (Line 9 to 10), as we are looking for UUs that, by definition, are incorrect predictions with high confidence.

Based on feature spaces comparison, we identify k problematic predictions that are most likely to be UUs and move these identified UUs into training data (Line 17) by collecting labels for them and replacing them in the test set by randomly adding new instances from the remaining images (Line 18). Then we re-train the model with this extended dataset, and repeat this process to discover more instances of unknown and under-represented classes until all classes are well represented in the training data. In the following, we provide detailed explanations of the functions used in Algorithm 1.

Algorithm 1: Iterative algorithm to discover UUs

```
Input: $D_0$ with unknown bias, entire image collection $D$
Parameter: $\tau$: confidence threshold,
           $k$: top-$k$ potential UUs to select iteratively
Output: balanced training set, prediction confidence
 1: Let $trainSet = D_0$, $testSet = \text{random}(D - D_0)$,
    $text = \varnothing$
 2: while $\neg$(all six classes are well represented) do
 3:     $model = \text{train}(trainSet)$
 4:     $\boldsymbol{p}\langle \boldsymbol{f}_{mac}, \boldsymbol{c} \rangle = \text{predict}_{model}(testSet)$
 5:     $text \leftarrow$ Read human input for $testSet$
 6:     $\boldsymbol{f}_{hum} = \text{extractFeature}(\text{aggregate}(text))$
 7:     Let $\rho_{rank} = \varnothing$
 8:     for $m \in testSet$ do
 9:         if $\boldsymbol{c}(m) < \tau$ then
10:             Skip $m$ and go to next
11:         else
12:             $\boldsymbol{sim}_{mac}(m) = \cos(\boldsymbol{f}_{mac}(m),\ \boldsymbol{f}_{mac}(n))$
13:             $\boldsymbol{sim}_{hum}(m) = \cos(\boldsymbol{f}_{hum}(m),\ \boldsymbol{f}_{hum}(n))$
                                        $(\forall n \in testSet \wedge n \neq m)$
14:             $\rho_{rank} \leftarrow \Gamma(\boldsymbol{sim}_{mac}(m), \boldsymbol{sim}_{hum}(m))$
15:         end if
16:     end for
17:     $trainSet \leftarrow \{m \mid m \in testSet \wedge$
                                $\rho_{rank}(m) \in \min_k(\rho_{rank})\}$
18:     $testSet = \text{random}(D - trainSet), text = \varnothing$
19: end while
20: return $\langle trainSet, \boldsymbol{p}\langle \boldsymbol{c} \rangle \rangle$
```

3.1 Generating Feature Spaces

To improve training performance, we use the popular pre-trained deep model ResNet50 (He et al. 2016) to extract deep features from the entire collection D. Then, we follow a random sampling strategy to select 1000 images with three classes equally represented as D_0, which has a bias obtained by removing training evidence for the other three classes. Next, we train an SVM (Hearst et al. 1998) classifier with D_0 and use this classifier to obtain the predicted labels and confidence scores on the test dataset (Algo. 1 Line 3 to 4). Note that other pre-trained deep models such as AlexNet (Krizhevsky, Sutskever, and Hinton 2012), VGG (Simonyan and Zisserman 2015) or ResNet101 (He et al. 2016) lead to similar results.

Since our goal is to identify objects in images, given the complexity of the images, we ask crowd workers to create a textual description for these images. To simulate the crowdsourcing task, we use the MS COCO captions dataset (Chen et al. 2015), which contains captions for each image generated by five independent human subjects over the Amazon MTurk platform. To extract the objects that the crowd mentions being present in the images, we adopt natural language processing (NLP) to analyze and aggregate their text (Algo. 1 Line 6). Based on the assumption that objects are primarily described by noun phrases, we first apply Part-of-Speech (POS) tagging to mark up each word (e.g., noun, verb, etc.) in these textual descriptions. Then, we extract two types of words: singular nouns (NN) and plural nouns (NNS), followed by plural to singular normalization

[2] An image may still contain multiple instances of that object.

[3] We maintain the size of test set at 1000 for the entire process.

and converting all characters into lower case. After this step, we obtain a set of nouns for each image representing the objects that have been mentioned by five different crowd workers. Considering the possibility of different workers using synonyms to describe the objects (e.g., airplane versus aeroplane), we then use WordNet (Miller 1995) to examine synonyms for each of these words, and pick up one as the "representative" word for each set of synonyms. Based on these representative words, we then count for each image their occurrence in the five text descriptions as a proxy for the frequency of object mentions by the five workers. Given the fact that these objects may sometime be described using super- or sub-concepts (e.g., bird versus pigeon, where "pigeon" is a "bird"), we again use WordNet to increase the frequency score of words at higher levels if workers mentioned it using a lower level word in the taxonomy. Specifically, if a word x appears in the path of other word y to the root r in the WordNet taxonomy (i.e., y is a subclass of x), then the frequency score of x is increased by the fraction of the length of the path from x to r divided by the length of the path from y to r, or $\frac{|level(x)-level(r)|}{|level(y)-level(r)|}$. With this method, we are able to rank objects identified by crowd workers to select the key one in each image.

After obtaining, for each image, a list of objects ranked by frequency, we adopt TF-IDF (Ramos et al. 2003) and adapt the formula to our use case to generate vectorial representations of the features extracted from the text generated by crowd workers (i.e., $\boldsymbol{f}_{hum}$, see Algo. 1 Line 6). For a given image m, we use the frequency (denoted by q_i) of each object (word) o_i divided by the sum of all frequency scores (i.e., $\sum_{\forall o_i \text{ for } m} q_i$) as its relative "term frequency" (TF) and define "inverse document frequency" (IDF) as the logarithm of the total number of images divided by the number of images that contain o_i. In this way, we are able to obtain a TF-IDF feature vector extracted from human generated text for each image (i.e., $\boldsymbol{f}_{hum}$).

3.2 Mining Unknown Unknowns

To understand how the human generated latent space differs from the machine learned feature space, we measure the similarity of the two spaces by looking at where similar images are placed. We take a given image m as query[4], and compute the cosine similarity of m with all other images by $\boldsymbol{f}_{mac}$ and $\boldsymbol{f}_{hum}$ (Algo. 1 Line 12 to 13). In this way we obtain two lists of images ranked by similarity scores using $\boldsymbol{f}_{mac}$ and $\boldsymbol{f}_{hum}$, respectively. Note that we focus on the ranking produced by the similarity scores rather than on their similarity values as we are interested in retrieving the most similar images (e.g., top-100 results) to the query image m. The two feature spaces (i.e., learned by the predictive model and from human input) can then be compared by computing the ranking correlation ρ_{rank} of the two ranked lists for each query image (Algo. 1 Line 14). For a query image m, a high $\rho_{rank}(m)$ value implies that the retrieved images are similar by either using $\boldsymbol{f}_{mac}$ or $\boldsymbol{f}_{hum}$. This indicates that $\boldsymbol{f}_{mac}(m)$ in the space learned by the predictive model is comparable to $\boldsymbol{f}_{hum}(m)$ in the space generated using human input. By contrast, a low $\rho_{rank}(m)$ value suggests that the most similar images retrieved by $\boldsymbol{f}_{mac}$ are different from those obtained by $\boldsymbol{f}_{hum}$, and thus the prediction of m may be problematic and likely to be a UU.

We adopt rank-biased overlap (RBO) (Webber, Moffat, and Zobel 2010) as our metric to compute ranking correlations (i.e., $\Gamma(\bullet)$, see Algo. 1 Line 14) due to its ability to (i) weight top ranked items more than the lower ranked ones (as we focus on top ranked similar images); (ii) handle incomplete rankings in a consistent way (as the query results may not have a full coverage of all similar images in the test set); and (iii) naturally adjust to any arbitrary cutoff (e.g., considering top-100 similar images per query as we do in our experiments). RBO measures the overlap between two ranked lists at incrementally increasing depths. Its value goes from 0 (completely different rankings) to 1 (exactly same rankings).

3.3 Expanding Training Set

Based on ρ_{rank}, we select k images[5] with the lowest ρ_{rank} and add their labels into the training set (Algo. 1 Line 17). We re-train the model with this extended dataset. To mimic a real scenario, we assign labels by aggregating the crowdsourced text for the selected images rather than the label which was originally removed from the training. We use the same method as extracting features of human input (Algo. 1 Line 6), and then are able to obtain the label for each image by picking up the word bearing the highest frequency score. To extend the train set, one way is to pick images for the three missing classes in D_0 (i.e., clock, airplane and giraffe). We call this method $Pick3$. Considering a more complex scenario in which problematic predictions with high confidence but low ρ_{rank} for known classes (e.g., predicting a striped bird to be a zebra) are also worth being added to the training set, we additionally consider picking images for all (i.e., six pre-defined) classes when expanding the training set. We call this method $Pick6$. Then, we re-train the model with the extended dataset and iteratively repeat the entire process to identify more instances of unknown and under-represented classes. We stop when all six classes are well represented in the training set (Algo. 1 Line 2)[6].

4 Results

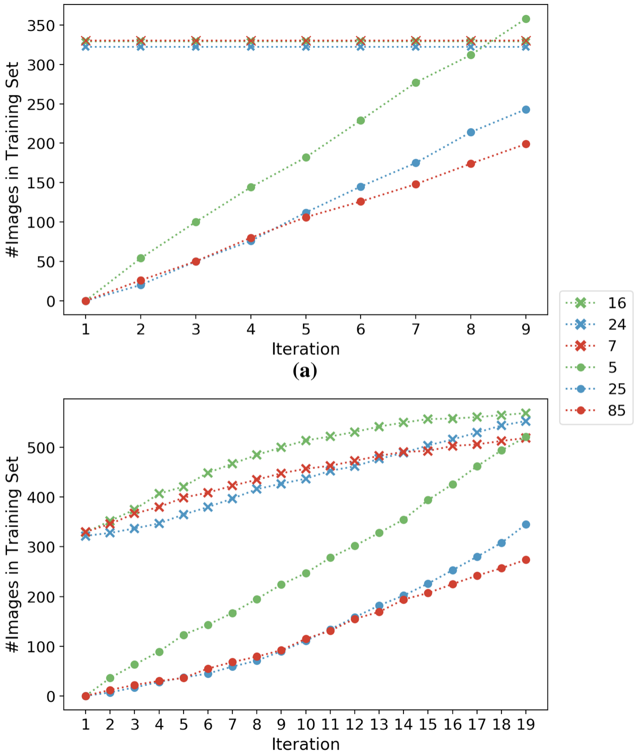
4.1 Identification of Unknown Unknowns

Figure 2 shows the number of images per class in the training set at each iteration. In the $Pick3$ approach (Fig. 2a), the number of images for all the three known classes remains

[4] Following previous work (Lakkaraju et al. 2017; Bansal and Weld 2018; Maurer and Bennette 2018), we set the confidence threshold $\tau = 0.65$, and only consider images whose prediction confidence is at or above this threshold.

[5] We set $k = 100$ in our experiment.

[6] The initial stopping criteria is based on the number of any UU class images exceeding the initial number (e.g., 300) of images for known classes. In our results (see Fig. 4) we also show that adding 200 images may lead to a comparable prediction quality as when using 300 images per class. Thus, such stopping criteria may be easily adjusted as required.



(a)

(b)

Figure 2: Number of images for each class in the training set over iterations while using (a) $Pick3$ and (b) $Pick6$. Notation: bird (16), zebra (24), train (7), airplane (5), giraffe (25), clock (85).

the same over the iterations (i.e., 329, 322 and 330 representing images of birds, zebras and trains, respectively). Instead, the number of images for the three unknown classes increases as we progress with more iterations. In the $Pick6$ approach (Fig. 2b), the number of images for all classes increases as more iterations are performed. This shows how algorithmic predictions can sometimes differ from human perception even if these classes are well represented in the training set. For example, some images identified as UUs are not really UUs, but are instead images of the other three known classes[7]. This shows that when our method picks the most problematic predictions (by comparing between human-data and machine-data generated spaces), the picked images may also include some instances of known classes (along with some for UU classes) due to the different positions of a specific image in the two generated spaces. This is because in the dataset, even though we only use images labeled as containing a single type of object, these images are indeed complex and may contain multiple objects (e.g., a bird on the water). Such complex images may be understood differently by the learning algorithm and by humans, and, thus, they may be positioned differently in the two spaces (i.e., human-data and machine-data generated spaces).

Iteration	**Known Classes**		**Unknown Classes**		**Total**
($Pick3$)	Correct	Wrong	Correct	Wrong	
1	479 (46.2%)	29 (2.8%)	—	529 (51.0%)	1037 (100%)
2	457 (46.3%)	28 (2.8%)	92 (9.3%)	410 (41.6%)	987 (100%)
3	453 (46.3%)	39 (4.0%)	224 (22.9%)	262 (26.8%)	978 (100%)
4	459 (46.5%)	37 (3.7%)	302 (30.6%)	189 (19.2%)	987 (100%)
5	453 (45.8%)	35 (3.5%)	371 (37.5%)	130 (13.2%)	989 (100%)
6	423 (44.5%)	39 (4.1%)	366 (38.5%)	122 (12.9%)	950 (100%)
7	468 (45.4%)	47 (4.5%)	410 (39.8%)	106 (10.3%)	1031 (100%)
8	445 (45.6%)	47 (4.8%)	391 (40.0%)	94 (9.6%)	977 (100%)
9	467 (44.7%)	59 (5.6%)	425 (40.6%)	95 (9.1%)	1046 (100%)

Iteration	**Known Classes**		**Unknown Classes**		**Total**
($Pick6$)	Correct	Wrong	Correct	Wrong	
1	479 (46.2%)	29 (2.8%)	—	529 (51.0%)	1037 (100%)
2	485 (48.0%)	27 (2.7%)	21 (2.1%)	476 (47.2%)	1009 (100%)
3	491 (47.9%)	30 (2.9%)	54 (5.3%)	450 (43.9%)	1025 (100%)
4	472 (46.5%)	29 (2.9%)	92 (9.0%)	423 (41.6%)	1016 (100%)
5	438 (47.5%)	24 (2.6%)	120 (13.0%)	340 (36.9%)	922 (100%)
6	450 (46.4%)	36 (3.7%)	176 (18.1%)	309 (31.8%)	971 (100%)
7	463 (46.6%)	32 (3.2%)	242 (24.3%)	258 (25.9%)	995 (100%)
8	455 (46.4%)	34 (3.5%)	283 (28.8%)	209 (21.3%)	981 (100%)
9	434 (45.5%)	30 (3.1%)	307 (32.2%)	183 (19.2%)	954 (100%)
10	447 (46.2%)	40 (4.1%)	346 (35.8%)	134 (13.9%)	967 (100%)
11	421 (45.6%)	39 (4.2%)	344 (37.2%)	120 (13.0%)	924 (100%)
12	438 (44.8%)	46 (4.7%)	379 (38.7%)	115 (11.8%)	978 (100%)
13	444 (45.6%)	45 (4.6%)	383 (39.4%)	101 (10.4%)	973 (100%)
14	425 (45.4%)	41 (4.4%)	372 (39.7%)	98 (10.5%)	936 (100%)
15	438 (45.5%)	49 (5.1%)	373 (38.8%)	102 (10.6%)	962 (100%)
16	442 (44.7%)	49 (4.9%)	415 (42.0%)	83 (8.4%)	989 (100%)
17	425 (44.8%)	49 (5.2%)	378 (39.8%)	97 (10.2%)	949 (100%)
18	404 (43.7%)	52 (5.6%)	377 (40.8%)	91 (9.9%)	924 (100%)
19	401 (42.9%)	53 (5.7%)	382 (40.9%)	98 (10.5%)	934 (100%)

Table 1: Number (and percentage) of predictions made over iterations with a breakdown by both prediction correctness (i.e., correct or wrong) and groundtruth (i.e., *known* classes covered by D_0 or *unknown* classes missing in D_0) using the $Pick3$ (top) and $Pick6$ (bottom) approaches.

Since the training data size for all six classes increases after each iteration by the $Pick6$ approach, this method requires more iterations (i.e., 19, see Fig. 2b) to have at least one unknown class (e.g., airplane) equally represented in the training set. Compared to this, the $Pick3$ approach only needs 9 iterations (Fig. 2a). It is evident that the number of discovered UUs varies from class to class. This shows that for some classes the problematic predictions are easier to identify while for other classes it is more difficult.

4.2 Quality of Algorithmic Predictions

To evaluate prediction quality we use precision, recall and F1 as our evaluation metrics for each of the six classes and compute the overall prediction accuracy over all classes at each iteration of the algorithm. For a given class u, a true positive (TP) is defined as a match between the predicted label v and the class, i.e., $v = u$. Otherwise, it is a false negative (FN) for class u and a false positive (FP) for class v. The overall prediction accuracy is defined as the number of correctly predicted labels divided by the total number of predictions.

Table 1 shows the number of correct and wrong predictions at each iteration using both $Pick3$ and $Pick6$ approaches. We group these predictions into four categories: (i) correct prediction of known classes (i.e., bird, zebra and train), (ii) wrong prediction of known classes, (iii) correct prediction of unknown classes (i.e., airplane, giraffe and clock), and (iv) wrong prediction of unknown classes. Since

[7] Note that we have in total six classes in our experiment: Three UU classes and three known classes.

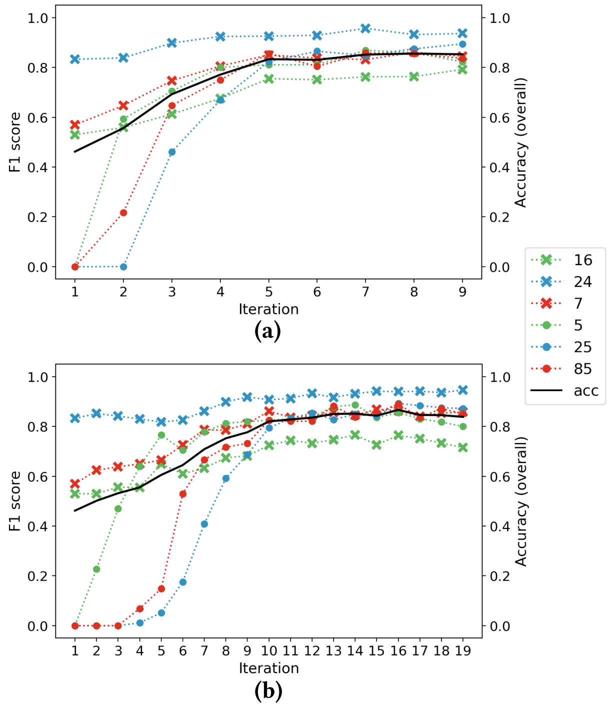


Figure 3: Quality of the predictions over iterations in terms of F1 score per class and overall accuracy, using (a) $Pick3$ and (b) $Pick6$ approaches. Notation: bird (16), zebra (24), train (7), airplane (5), giraffe (25), clock (85), overall accuracy (acc).

we manually remove three (out of six) classes in D_0, the model is not able to correctly predict the three unknown classes at the first iteration. As we progress with the iterations, the number of correct predictions for the three unknown classes increases gradually using both methods, suggesting that by adding training instances for unknown classes (see Fig. 2) the model performance is improved. Figure 3 shows F1 scores and overall accuracy values for the predictions using both $Pick3$ and $Pick6$. Since half of the true labels are missing from the training set D_0, the overall accuracy of the prediction in the first iteration is 0.46 for both methods as the three unknown classes cannot be correctly predicted by the model. With more iterations being performed, the overall accuracy increases going up to 0.83 in iteration 5 using $Pick3$ (Fig. 3a) and up to 0.82 in iteration 10 using $Pick6$ (Fig. 3b). The overall accuracy for both methods does not improve much afterwards.

In terms of F1 scores, three unknown classes (i.e., airplane, giraffe and clock) get 0 in the first iteration because the predictive model cannot recognize them. As more iterations are performed, the F1 scores for these classes increase significantly until the fifth and tenth iterations for $Pick3$ (Fig. 3a) and $Pick6$ (Fig. 3b) and then remain at the same levels, respectively. This explains why the overall accuracy does not improve much afterwards. On the contrary, F1 scores in the first iteration for the three known classes are 0.53, 0.83 and 0.57 for bird, zebra and train, respectively.

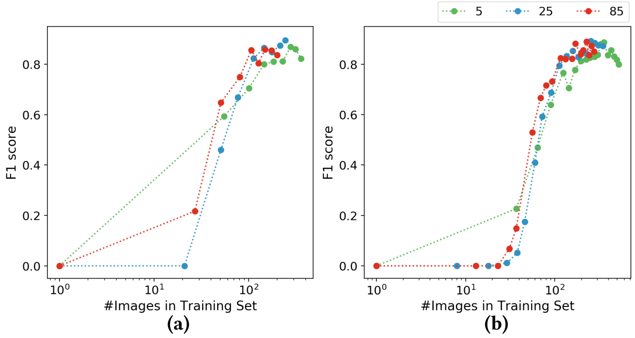


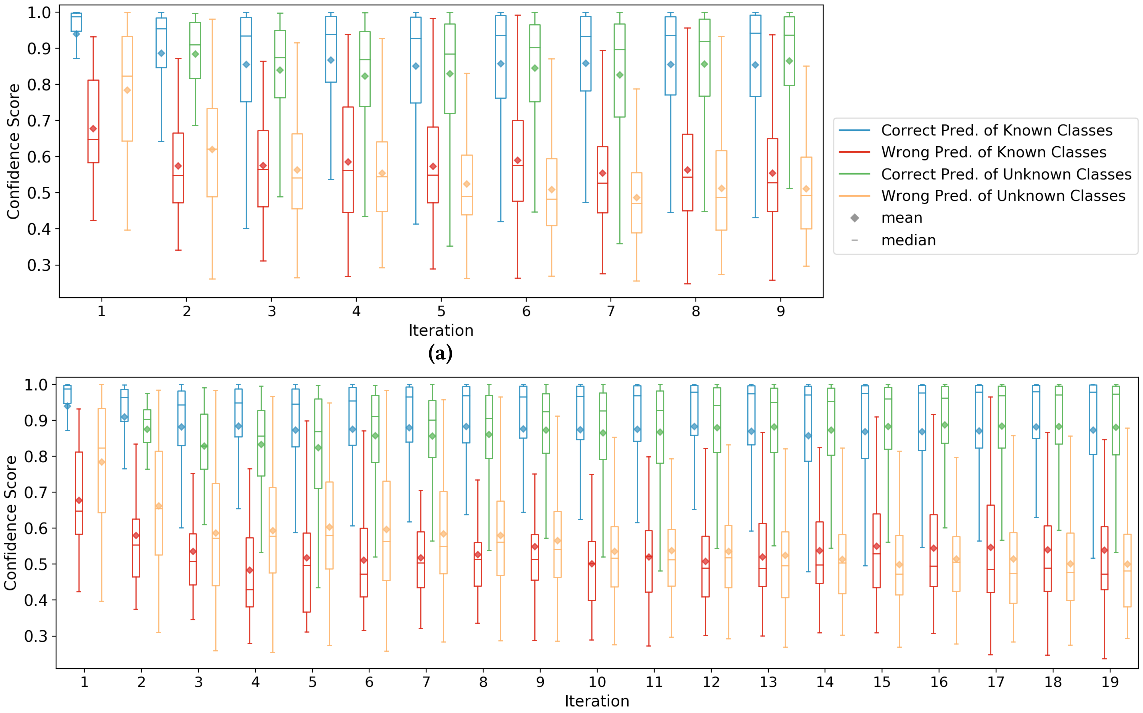
Figure 4: F1 scores for the three unknown classes over the number (in log scale) of images included in the training data, using (a) $Pick3$ and (b) $Pick6$. Notation: airplane (5), giraffe (25), clock (85).

This shows how the learning algorithm has extracted some features that make zebra images well distinguishable from other classes, and thus most UUs are images classified as bird or train rather than zebra.

Among the three unknown classes (i.e., airplane, giraffe and clock), the improvement of the prediction quality for airplane happens at a higher rate than for clock or giraffe. This is because there are more airplane images identified as UUs and subsequently added into the training set after each iteration (see Fig. 2). This observation implies that the features learned by the predictive model for airplane images show the largest deviation from human perception. This makes the airplane class be the first to get an equal train set size as the three known classes (see Fig. 2). On the other hand, due to the complexity of the images (e.g., presence of multiple objects), adding too many examples to the training set does not necessarily improve the quality of the predictions (e.g., the F1 scores for both airplane and bird decrease after iteration 16 by $Pick6$ approach, see Fig. 3b) or can even increase the number of wrong predictions of the known classes (see Tab. 1). Figure 4 shows the improvement of the prediction quality by means of F1 scores for the three unknown classes with respect to the number of images included in the training data. We can observe that by adding 100 images per class, the F1 score can reach high values (around 0.8) and does not increase much as we add more than 200 images for each class. This can be achieved in five iterations using $Pick3$ and in ten iterations using $Pick6$ (see Fig. 2). This bears implications on the decision of when to stop with iterations (Algo. 1 Line 2) in order to obtain the most cost-effective benefits (see Sec. 3.3).

4.3 Effects on Model Confidence

Next, we look at how the *confidence* of the predictions changes over iterations. Recall that UUs are defined as incorrect predictions with high confidence scores. To understand how the confidence scores for the predictions of *unknown* classes (i.e., airplane, giraffe and clock) differ from those for the predictions of *known* classes (i.e., bird, zebra and train), we group them by the correctness of the predic-



(a)

(b)

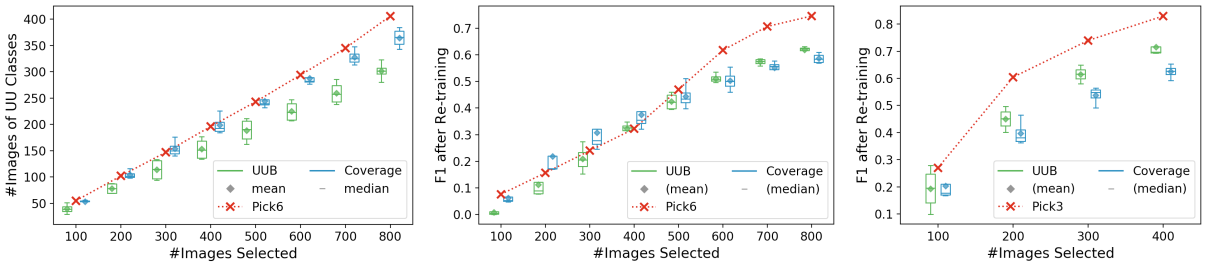
Figure 5: Confidence of the predictions over iterations by (a) $Pick3$ and (b) $Pick6$ approaches. The predictions are grouped into four categories: (i) correct prediction of known classes (i.e., bird, zebra and train), (ii) wrong prediction of known classes, (iii) correct prediction of unknown classes (i.e., airplane, giraffe and clock), and (iv) wrong prediction of unknown classes.

tions (i.e., correct or wrong) and the groundtruth labels (i.e., known or unknown classes). Figure 5 shows the evolution of the confidence scores over iterations with a breakdown in the four groups: (i) correct prediction of known classes, (ii) wrong prediction of known classes, (iii) correct prediction of unknown classes, and (iv) wrong prediction of unknown classes. At the first iteration, due to the absence of the three unknown classes in D_0, all the predictions for these classes are incorrect (see Tab. 1). However, the confidence scores of these wrong predictions (mean 0.785 and median 0.823) are higher than those of the wrong predictions for known classes (mean 0.678 and median 0.649) in this iteration (Fig. 5ab). Using a Mann-Whitney U test[8], we conclude that this difference is statistically significant ($p < 0.01$). This confirms the existence of UUs, as the model assigns high confidence scores to wrong predictions.

As more images of unknown classes are being added to the training set, in the second and third iterations (see Fig. 2ab) the confidence score of wrong predictions for unknown classes drops significantly (Fig. 5ab) (Mann-Whitney U test reveals statistically significant difference, $p < 0.01$). This shows that by adding some (but not enough) labels for unknown classes and re-train the model, the prediction uncertainty for these classes increases when they are incorrectly predicted. This suggests that the predictive model can, to some extent, realize that mistakes are being made just after three iterations. As we progress doing more iterations, the confidence of wrong predictions for unknown classes decreases to a comparable level to that of wrong predictions for known classes at Iteration 2 and Iteration 9 using $Pick3$ (Fig. 5a) and $Pick6$ (Fig. 5b), respectively. We do not observe statistically significant difference between the two groups. Meanwhile, the confidence of correct predictions for unknown classes is higher than that of wrong predictions for these classes in all iterations (i.e., from Iteration 2) using both $Pick3$ and $Pick6$ approaches (Mann-Whitney U test, $p < 0.01$). These observations show that the performance of the model in predicting unknown classes (as either correct or wrong predictions) is similar to that in predicting known classes (i.e., wrong predictions being assigned lower confidence scores) after introducing a certain number of labels for unknown classes into the training set. The discriminative power of the model in distinguishing (potentially) wrong predictions from correct ones manifests that UUs have been pushed into *known* unknowns, where the mistakes made by the model are able to be identified by the low confidence.

Another observation we can make is that the prediction confidence decreases in the first few iterations of correct predictions for both known and unknown classes, as well as those of wrong predictions for unknown classes. This can be explained by the increased diversity of images (e.g., different objects with the same background) when the training set

[8] We adopt the Mann-Whitney U test (Mann and Whitney 1947) to examine the differences among these scores across iterations as the score distribution is not interval scaled.



(a) (b) (c)

Figure 6: Comparison with baseline methods: (a) number of identified UUs and (bc) F1 of the model prediction after retraining.

expands. When we add more images for unknown classes to the training set in subsequent iterations, the model confidence for correct predictions goes gradually up again. Confidence scores of correct predictions for unknown classes become similar to those for known classes from Iteration 7 and Iteration 15 onward using $Pick3$ (Fig. 5a) and $Pick6$ (Fig. 5b) respectively. This suggests that after learning from a more balanced training set where all classes are better-represented (see Fig. 2), the predictive model is able to make more reliable predictions.

4.4 Comparison with Baseline Approaches

In both UUB (Lakkaraju et al. 2017) and coverage-based methods (Bansal and Weld 2018), the instances (i.e., images in our dataset) in the test set are selected and used to query human subjects one by one and the corresponding human label (regarded as the groundtruth) is used to make a decision on selecting the next instance (or cluster) to query human subjects for. Thus, we compare the proposed method with these two human-in-the-loop approaches by assuming that a certain number (e.g., 100) of instances are required to be selected. We evaluate selected instances from two perspectives: (i) the number of true UUs among selected images, and (ii) the quality (i.e., F1 averaged across all UU classes) of the model prediction after we add all selected images to the training set and retrain the model.

Figure 6 presents the comparison between our methods and baseline approaches[9]. We can see that our method (i.e., $Pick6$) outperforms the baseline approaches in terms of the number of identified UUs (Fig. 6a). In terms of model prediction quality after retraining (see Fig. 6b), as coverage-based approach is able to select images that have a better coverage of the three UU classes, it exhibits a better improvement of average prediction quality than our approach while selecting up to 500 images to retrain the model. Indeed, our method works iteratively and tends to select instances that bear the highest dissimilarity between human and machine generated spaces. Thus, the images selected by our method may come from the classes having the most problematic predictions, and, for this reason, these images may not have a balanced coverage of all UU classes. On the other hand, as more instances (e.g., more than 500 images) are being selected by our iterative method and the coverage of all UU classes is improved, our method shows better performance than the baseline approaches. If the existence of UUs is known a priori (i.e., we can directly pick instances of UU classes to retrain the classification model), our $Pick3$ method exhibits better performance than the baseline approaches (see Fig. 6c). This indicates that by selecting the most problematic images, our method is able to improve the model performance (based on F1 score) more efficiently as compared to the other methods in terms of the required instances with human generated labels.

5 Discussion

Through the experimental results presented in the previous section, we have shown that our human-in-the-loop approach is effective in the discovery of UUs and can improve the quality of the prediction model by iteratively selecting new instances to be labelled. Our proposed method can be generalized to address the UU detection problem that may be caused by other reasons like, for example, covariate shifts (Sugiyama and Kawanabe 2012). As the method looks at the prediction results, it collects manual labels for selected instances from the test set, and finally adds these selected instances to expand the training data. If the UU problem is caused by distributional differences between training and test data, for example, adding new data points from the test set would gradually address this problem. Thus, our method is also effective to address such UU causes.

In our experiments, we show that it is possible to effectively identify the most problematic predictions by feature space differentiation. Measuring the distance of each image to other images allows us to understand how the relations among these images differ from one space to another. In this way, we are able to identify which images have different representations in the machine generated space, compared to the human generated space. Then, adding these images to the training set results into better quality predictions. This bears implications on understanding the machine learning black-box from a human point of view (Krause, Perer, and Ng 2016) and, in turn, on building more human-like algorithms to process multimedia content, for example. In our experiments, some complex images may result in positioning differences across the human and machine generated spaces. For this reason, some images from known classes may also be picked up by our method in each iteration (if the image is

[9] We execute the baseline methods 10 times to reduce the impact of the random seeds used in their initialization steps. This is reflected by the boxplots in Figure 6.

complex). In Iteration 1 of our $Pick6$ method (see Fig. 6a), for example, of the 100 images that are picked up, 55 (out of 100) images are from UU classes (i.e., the other 45 images of from known classes)[10]. On the other hand, our method is not restricted to working on image data as it uses the human-data generated space to differentiate the machine-data generated space and it assumes the human generated data as groundtruth. As our method relies on feature space differentiation, it is independent of the types of data in which UU problems are present. As humans are able to generate text (e.g., writing comments) for different types of data like videos, so it is possible to construct the two required feature spaces we introduce in this work. Therefore, our iterative human-in-the-loop approach is directly applicable to other scenarios, such as identifying UUs in textual content.

An iterative approach can also provide a more cost-effective UU detection method. In the proposed methodology, we initially set the stopping condition as having all classes well represented in the training data (Algo. 1 Line 2). The prediction quality in our experiments, however, indicates that having 100 (and no more than 200) images per class to train the model is enough to obtain good quality predictions (see Fig. 4). Depending on the characteristics of the task (e.g., the complexity of the images to be classified), we may even need to perform fewer iterations to achieve good quality predictions (e.g., we only need five rather than ten iterations of $Pick6$ to get good predictions of "airplane" images, see Fig. 3b). On the other hand, adding more images to the training set allows to make more reliable predictions, as the (median) confidence scores for correct predictions for both known and unknown classes increase over iterations (see Fig. 5). Such findings are valuable in a crowdsourcing setting and allow for new stopping criteria to be defined to provide the most cost-effective trade-offs. Since querying crowd workers is expensive, we do not want to access extra human work if it does not contribute to improving prediction quality given that certain reliability is achieved.

Moreover, our iterative approach shows its ability to make UUs become *known unknowns*, where incorrect predictions come with low confidence scores. In our experiment, in the first iteration the predictive model makes high confidence predictions for unknown classes (see Fig. 5) but these predictions are actually wrong (see Tab. 1), as they are UUs (i.e., errors which are blind to the model). After performing two more iterations, the model confidence of wrong predictions for unknown classes drops significantly. This shows that by adding discovered UUs to the training data, the predictive model becomes aware of uncertain predictions that are likely to be wrong. As more iterations are performed, the model confidence of wrong predictions becomes distinguishable from that of correct predictions for both known and unknown classes. Thus, these potentially incorrect predictions can now be automatically identified using their confidence scores being below a certain threshold. Because of the existence of UUs, an uncertainty sampling strategy (Lewis and Catlett 1994) does not work as UU instances do not come with low confidence scores. By pulling down the confidence scores of wrong predictions, we are able to make the uncertainty sampling strategy work again. This bears implications on building hybrid human-machine systems to efficiently discover UUs with a constrained manual annotation budget. For example, by asking humans to identify a certain amount of UUs with just a few iterations, we are able to push UU instances into known unknowns, and then use uncertainty sampling strategy to automatically select problematic predictions for manual post-processing. Such a two-stage process would benefit AI practitioners in cutting costs or in making the most of their budget to carry out experiments and build products.

6 Conclusions

Automatic prediction models may result in wrong but high confidence classification decisions. This usually happens when certain classes are absent or under-represented in the training data. In this paper, we present the first work that leverages human intelligence in an active learning fashion to discover unknown unknowns. By comparing the feature space generated by human input from that learned by the predictive model from images, we iteratively discover the unknown and under-represented classes in the training data and extend it to re-train the model.

The experimental results show that the proposed method is able to significantly improve model prediction quality, as well as to effectively push UUs to known unknowns by reducing the model confidence for UU instances so that they may be automatically selected for post-processing after initial human intervention. We also provide insights of the trade-off between the prediction quality improvements and the human efforts required to achieve those improvements. These results bear implications on the design of cost-effective and task-dependent systems to discover unknowns with humans in the loop.

Our method relies on space differentiation that looks at how instances are positioned in each (human- or machine-generated) space. This is computed by ranking each instance relative to other instances in the space. In the extreme case where all instances are identical and indistinguishable in the representational space, our method would not be able to detect UU instances, which is a limitation. This case, however, is not likely to occur in practice as in real-world scenario instances are usually modeled by multi-dimensional representations (e.g., image embeddings) and instances being identical would require all the features being the same. In fact, our method can generate effective rankings as long as at least one feature dimension is different for different instances, making our approach applicable across data types.

Acknowledgments

The image caption dataset is created in the MS COCO project (Lin et al. 2014; Chen et al. 2015). This work is partially supported by an ARC Discovery Project (No. DP190102141), an ARC ITTC Project (No. IC200100022), and by a Google AI for Social Good award.

[10] Note that the method picks 100 images per iteration, where some images from known classes are also picked up by other methods (including baseline approaches, see Fig. 6a).

References

Andriluka, M.; Uijlings, J. R.; and Ferrari, V. 2018. Fluid annotation: a human-machine collaboration interface for full image annotation. In *Proceedings of the 26th ACM international conference on Multimedia*, 1957–1966.

Attenberg, J.; Ipeirotis, P.; and Provost, F. 2015. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *Journal of Data and Information Quality (JDIQ)* 6(1): 1.

Bansal, G.; and Weld, D. S. 2018. A coverage-based utility model for identifying unknown unknowns. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* .

Cheng, J.; and Bernstein, M. S. 2015. Flock: Hybrid crowd-machine learning classifiers. In *Proceedings of the 18th ACM conference on Computer Supported Cooperative Work & social computing*, 600–611. ACM.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hearst, M. A.; Dumais, S. T.; Osuna, E.; Platt, J.; and Scholkopf, B. 1998. Support vector machines. *IEEE Intelligent Systems and their applications* 13(4): 18–28.

Jing, X.-Y.; Wu, F.; Li, Z.; Hu, R.; and Zhang, D. 2016. Multi-label dictionary learning for image annotation. *IEEE Transactions on Image Processing* 25(6): 2712–2725.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. 2009. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2372–2379. IEEE.

Kovashka, A.; Vijayanarasimhan, S.; and Grauman, K. 2011. Actively selecting annotations among objects and attributes. In *2011 International Conference on Computer Vision*, 1403–1410. IEEE.

Krause, J.; Perer, A.; and Ng, K. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 5686–5697.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 1106–1114.

Kulesza, T.; Stumpf, S.; Burnett, M.; Wong, W.-K.; Riche, Y.; Moore, T.; Oberst, I.; Shinsel, A.; and McIntosh, K. 2010. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, 41–48. IEEE.

Lakkaraju, H.; Kamar, E.; Caruana, R.; and Horvitz, E. 2017. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Lewis, D. D.; and Catlett, J. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, 148–156. Elsevier.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, A.; Guerra, S.; Fung, I.; Matute, G.; Kamar, E.; and Lasecki, W. 2020. Towards Hybrid Human-AI Workflows for Unknown Unknown Detection. In *Proceedings of The Web Conference 2020*, 2432–2442.

Mann, H. B.; and Whitney, D. R. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* 50–60.

Maurer, K.; and Bennette, W. 2018. Facility Locations Utility for Uncovering Classifier Overconfidence. *arXiv preprint arXiv:1810.05571* .

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11): 39–41.

Nushi, B.; Kamar, E.; and Horvitz, E. 2018. Towards accountable AI: Hybrid human-machine analyses for characterizing system failure. In *Sixth AAAI HCOMP Conference*.

Nushi, B.; Kamar, E.; Horvitz, E.; and Kossmann, D. 2017. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramakrishnan, R.; Kamar, E.; Dey, D.; Shah, J.; and Horvitz, E. 2018. Discovering blind spots in reinforcement learning. In *17th AAMAS Conference*, 1017–1025.

Ramos, J.; et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, 133–142.

Reiter, R. 1981. On closed world data bases. In *Readings in artificial intelligence*, 119–140. Elsevier.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM transactions on graphics (TOG)* 23(3): 309–314.

Siddiquie, B.; and Gupta, A. 2010. Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2979–2986. IEEE.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

Sroka, J. J.; and Braida, L. D. 2005. Human and machine consonant recognition. *Speech Communication* 45(4): 401–423.

Sugiyama, M.; and Kawanabe, M. 2012. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Vandenhof, C. 2019. A Hybrid Approach to Identifying Unknown Unknowns of Predictive Models. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, 180–187.

Venugopalan, S.; Anne Hendricks, L.; Rohrbach, M.; Mooney, R.; Darrell, T.; and Saenko, K. 2017. Captioning images with diverse objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5753–5761.

Webber, W.; Moffat, A.; and Zobel, J. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)* 28(4): 20.

Wu, B.; Chen, W.; Sun, P.; Liu, W.; Ghanem, B.; and Lyu, S. 2018. Tagging like humans: Diverse and distinct image annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7967–7975.

Yang, J.; Smirnova, A.; Yang, D.; Demartini, G.; Lu, Y.; and Cudré-Mauroux, P. 2019. Scalpel-CD: leveraging crowdsourcing and deep probabilistic modeling for debugging noisy training data. In *The World Wide Web Conference*, 2158–2168. ACM.

Yao, A.; Gall, J.; Leistner, C.; and Van Gool, L. 2012. Interactive object detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3242–3249. IEEE.

Zhang, Z.; Singh, J.; Gadiraju, U.; and Anand, A. 2019. Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction* 3(CSCW): 1–23.