Image classification using class-agnostic object detection[⋆]

Geoffrey Holmes[0000−0003−0433−8925], Eibe Frank[0000−0001−6152−7111], and Dale Fletcher[0000−0002−0455−8744]

University of Waikato, Hamilton, New Zealand
geoff,eibe,dale@waikato.ac.nz

Abstract. Human-in-the-loop interfaces for machine learning provide a promising way to reduce the annotation effort required to obtain an accurate machine learning model, particularly when it is used with transfer learning to exploit existing knowledge gleaned from another domain. This paper explores the use of a human-in-the-loop strategy that is designed to build a deep-learning image classification model iteratively using successive batches of images that the user labels. Specifically, we examine whether class-agnostic object detection can improve performance by providing a focus area for image classification in the form of a bounding box. The goal is to reduce the amount of effort required to label a batch of images by presenting the user with the current predictions of the model on a new batch of data and only requiring correction of those predictions. User effort is measured in terms of the number of corrections made. Results show that the use of bounding boxes always leads to fewer corrections. The benefit of a bounding box is that it also provides feedback to the user because it indicates whether or not the classification of the deep learning model is based on the appropriate part of the image. This has implications for the design of user interfaces in this application scenario.

Keywords: Human-in-the-loop Machine Learning, Convolutional Neural Networks, Image Classification, Object Detection

1 Introduction

Image classification using deep convolutional networks is one of the most prominent practical applications of machine learning. The learning algorithms for these networks require labelled images as training data. Often, obtaining these labels requires access to sophisticated domain expertise (e.g., in the medical domain), which can be costly. Thus, it is important to provide mechanisms to obtain correct labels for images in the most efficient manner possible.

It is important to note that this applies even when the total amount of labelled data required can be reduced by applying transfer learning so that the

[⋆] This work is part of a user-friendly deep learning project funded by the New Zealand Ministry for Business, Innovation and Employment.

learning of a neural network does not have to start with random parameter settings. For example, a standard strategy for transfer learning in convolutional neural networks is to take a network that has been pre-trained on a large collection of images, such as the well-known ImageNet database consisting of more than a million images, each furnished with one of 1,000 class labels, and fine-tune the parameters of this network on the labelled target domain data that is available once the classification "head" of the network has been replaced to feature as many classes as are present in the target domain. Even though this standard form of transfer learning can dramatically decrease the amount of labelled data required to achieve a satisfactory level of accuracy, there generally remains a substantial amount of labelling effort that must be applied to obtain a sufficient amount of labelled data for the target domain. Hence, even if transfer learning is applied—and we do apply it in this paper—a procedure for efficiently generating correct labels for this data is very useful.

A central idea in this context is to apply a form of human-in-the-loop machine learning, where the expert provides a small initial set of labelled examples for training a classifier, which is subsequently applied to *pre*-label batches of unlabeled data before they are passed for inspection to the expert, who then simply needs to *correct* the provided labels rather than determining labels from scratch for unlabeled images. Crucially, a proxy for the human effort required in this process is the number of *corrections* that the expert must perform, not the total number of examples to be labelled.

Earlier work [5] has investigated how best to order the batches of unlabeled data to minimise the number of corrections that need to be performed to train the image classifier to a satisfactory level. A key outcome of this work is that so-called "active" learning strategies [10] (e.g., uncertainty sampling) are inappropriate when the desired outcome is to minimise this measure of required effort: random example ordering generally yields a lower number of required corrections regardless of the image classification dataset and neural network architecture applied. Another strategy that performed well in [5] is to select a representative sample of unlabeled data *a priori* (i.e., before labelling/learning starts) using an algorithm called kernel herding [2], which attempts to improve on random sampling by ensuring good coverage of the full population. Intuitively, the poor performance of active learning can be understood by considering that it is based on selecting those examples for labelling that "surprise" the classifier the most (e.g., by considering its predictive uncertainty). These are clearly often examples where the predicted label is incorrect and must therefore be corrected by the human involved.

In this paper, building on these findings, we investigate whether the labelling effort can be further reduced by enabling the expert to provide additional information to speed up machine learning in the human-in-the-loop system. More specifically, we consider whether the use of a bounding box that the expert draws around the part of the image that is deemed responsible for its class label can help to reduce the number of corrections that this expert must perform during the learning process—focusing again on the number of times that a predicted class label must be corrected in the human-in-the-loop process, *not* the total number of class labels required. Two strategies of supplying bounding boxes are applied in conjunction with the two *a priori* sampling methods from the previous study—random sampling and kernel herding—to determine whether the use of this additional information can assist in the overall goal of reducing the number of corrections a user has to make. With both strategies, a standard object detection model is trained in a class-agnostic manner based on the bounding boxes provided by the expert. The expert-provided or, when available, a predicted bounding box is used to extract the part of the image that is passed to the image classification model to enable the association with a class label. This is compared to the set-up from [5], where the entire image is associated with a label.

2 Related Work

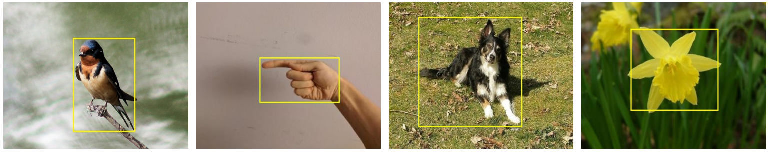
There is a vast amount of literature on human-in-the-loop object detection that looks at how interactive machine learning can reduce the number of annotations such as bounding boxes that the human must provide to enable learning of an accurate object detection model. Some of this work is surveyed in [11]. It is important to note that this is not the problem that is the focus of this paper. Instead, we apply object detection as a tool to improve the efficiency of human-in-the-loop learning for image classification.

Our use of an object detection method can be viewed as a form of class-agnostic object detection, a concept that has been introduced fairly recently in the literature in [7]. According to [7], in class-agnostic object detection, “the goal is to predict bounding boxes for all objects in an image but not their object-classes”. We focus on the special case where a single object is present in the image and the detector provides a single bounding box for this object. This bounding box is subsequently used to crop the image before it is passed to the learning algorithm for the image classification model.

Related to our work are approaches that attempt to use auxiliary information to improve the accuracy of image classification. [8] consider the use of bounding boxes as auxiliary information and provide a learning algorithm and neural network architecture that can exploit this information, looking primarily at whether this improves the quality of the explanations of predictions given in the form of saliency maps because accurate explanations are important for establishing trust.

Interestingly, [6] finds that enabling users to provide feedback to a simulated object detection system by providing the ability to correct bounding boxes based on the human-in-the-loop process *lowers* their trust in the machine learning system, regardless of whether the accuracy of object detection improves or not. Considering the potential use of object detection in safety-critical applications, this quite plausibly more realistic assessment of the algorithms’ ability may be appropriate.

Fig. 1. Ground truth bounding boxes



We are unaware of any work in the literature that investigates the use of class-agnostic object detection to improve the efficiency of human-in-the-loop machine learning for image classification.

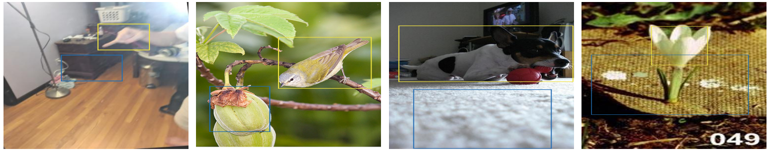
3 Set-up for the experiments

For each of the image classification datasets considered in our experiments, we start with a pre-trained image classification model that has been trained on ImageNet. Subsequently, data is provided to the learning algorithms in batches. Assuming the base case considered in [5], where no object detection is performed, only labels need to be provided for the images in a batch. For the first batch, the human in the loop is required to label all the images in this batch. Subsequently, the pre-trained image classification model is fine-tuned on this batch (details are given below) and applied to pre-label the images in the next batch, which the user must then (potentially) correct. Once the labels have been finalized, the pre-trained model is fine-tuned on the extended dataset, comprising both the first and the second batch of data, before the third batch of data is processed in the same manner. This is repeated until a sufficient level of accuracy is achieved or the data is exhausted. We examine two approaches to ordering training images into batches: random selection and kernel herding (both are described in [5]). Both approaches can be applied before the loop starts because they do not require knowledge of any class labels.

This base case approach to human-in-the-loop learning for image classification does not apply object detection and does not use bounding boxes. We introduce bounding boxes into the process and investigate whether this improves efficiency.

Figure 1 shows some ground-truth bounding boxes for the datasets used in the experiments. These bounding boxes must also be generated by using the human in the loop. However, once at least some of the training data has been furnished with bounding boxes, one can train a standard object detection model on these boxes. In this paper, we build a model of the bounding boxes from the training data for each dataset using Faster RCNN [9] by applying a ResNet101 base network. This was chosen simply because it is popular and effective. The model is trained in a class-agnostic manner: it is simply configured to detect "the object" in the image, regardless of the class of the object. As the model can output multiple bounding boxes for an image, we simply pick the one for which the model exhibits the greatest confidence.

Fig. 2. Model misunderstandings



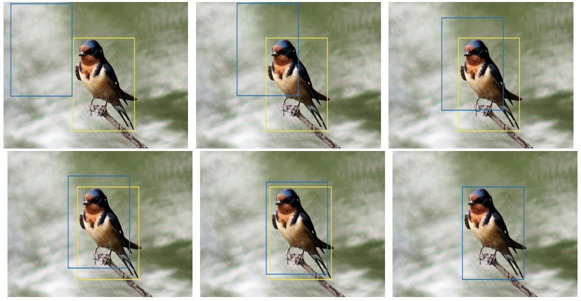
Once trained, the model can be used to predict bounding boxes for new images. It is important to note that this can happen as soon as the first batch of data has been annotated with boxes (and class labels) in the human-in-the-loop process, and the object detection model has been trained on this batch. Thus, in the second and subsequent batches, we can use the object detection model to predict bounding boxes, which the user may or may not correct—we investigate both approaches in the experiments.

In many cases, bounding boxes can explain why classification fails. For illustration, Figure 2 shows predicted bounding boxes, in blue, for images where the ground truth is also shown in yellow. It should be noted that each of these images involves the correct classification of the class label by the corresponding image classification model. However, given the blue bounding boxes, the reason for the classification is erroneous. The object detection model predicts, for the leftmost image, a dark area of a small chest of drawers instead of the hand. The next image to the right is easier to understand, as the bird does share a lot with the fruit in the predicted bounding box. The confusion in the dog image, between the carpet and the dog, may well be due to the similarity of the white part of the dog and the carpet. For the final image, there is some overlap between the flower and the predicted area, which does at least contain the stem and some of the petals of the flower.

The last image illustrates why a measure of overlap between predicted and ground-truth bounding boxes is used to evaluate object detectors. The intersection over union (IoU) provides such a measure, and Figure 3 shows the relationship between two boxes based on this measure as the degree of overlap moves from 0.0 (no overlap in the top left image) to 1.0 (fully overlapped in the bottom right). Object detection methods are commonly evaluated using this measure by considering IoU values above a certain threshold (e.g., 0.5) as a match. Faster-RCNN with a ResNet101 backbone is known to yield high performance according to this metric (see, e.g., [1]) and this is why it is used in our experiments.

The predicted bounding boxes for the second and subsequent batches of data can be used to crop the images so that they can be fed to the image classifier. In line with the above discussion, our experiments in this paper involve two update strategies for the object detection model. One method simulates the user correcting the bounding boxes every iteration (referred to as BB) and the other simulates the user only providing correct bounding boxes for the first batch of data (referred to as BB-1). The latter strategy aims to determine the effectiveness of a minimalist approach, where the user only needs to correct the bounding

Fig. 3. Intersection over Union in increments 0.0, 0.2, 0.4 on the top row from left to right, and 0.6, 0.8, 1.0 on the bottom row from left to right



boxes available for the very first batch of data, and the object detection model remains fixed afterwards throughout the human-in-the-loop process.

Whether corrected or uncorrected bounding boxes are used, the bounding box generated for an image is used to crop the image before it is passed to the image classifier to obtain a label. The image classification model—we evaluate pre-trained MobileNet, ResNet50, and ResNet152 models—is fine-tuned on the cropped training dataset, and evaluated against the cropped test and validation datasets. For the test and validation datasets, the cropping is based on the predicted bounding boxes produced by the current state of the bounding box prediction model: corrected boxes are obviously not available when the trained image classification system—incorporating both the object detection model and the image classifier—is deployed in practice, and thus measures of predictive performance must be based on predicted bounding boxes only.

3.1 Dataset Descriptions

We conducted experiments using the four publicly available datasets listed in Table 1. These datasets exhibit various collection sizes (ranging from 1300 to 20,000), class counts (17 to 200), and modeling complexity (easy to difficult). All datasets contain bounding box annotations, which are rectangles around the object of interest that identify the image's label. Whenever a "corrected" bounding box is required in our experiments, we use the corresponding ground-truth box provided by the annotations: we assume that the user makes perfect corrections in the simulated human-in-the-loop process we use in our experiments. This applies to both bounding boxes and labels. Ground-truth bounding boxes are shown in yellow for examples from each dataset in Figure 1.

The 17Flowers dataset[1] contains 80 images per class with varying backgrounds, but each image displays a centrally placed flower.

The American Sign Language dataset[2] comprises 50 to 90 images per label. Each image depicts a hand forming the sign of a single letter of the English al-

[1] https://www.robots.ox.ac.uk/ vgg/data/flowers/17/

[2] https://public.roboflow.com/object-detection/american-sign-language-letters

Table 1. Datasets

Name	Number of Examples	Number of Classes
17Flowers	1360	17
American Sign Language	1728	26
Stanford Dog Breeds	20,580	120
Birds	11,788	200

phabet, and the hand appears in the same location in each image under relatively consistent lighting conditions, rendering this dataset relatively uniform.

The Dog Breeds dataset[3] contains 120 classes with between 100 to 200 images per class, classified by breed. The position of the dog and the background differ substantially in this dataset.

The Birds dataset[4] contains 200 categories, most of which include 59 or 60 images, with some categories having fewer, and the smallest class containing 41 images. Each image shows a single bird, classified by species, with the bird typically positioned centrally, but with varying backgrounds.

3.2 Methodology

Our experimental methodology aims to simulate the human-in-the-loop training process. We implement our experiments in Python, utilizing the Keras deep-learning library, which provides pre-trained models [3]. We chose three pre-trained models: Mobilenet, Residual Networks [4] (ResNets) with 50 layers, and ResNets with 152 layers. Mobilenet is preferred for lightweight applications, particularly for mobile applications, while ResNets are typically used in medium to heavyweight applications. We use Docker images to control the GPU environment (Nvidia GeForce GTX 1080Ti).

The experimental process consists of generating a stratified holdout dataset comprising 15% of the images from each class at random. This dataset is used to evaluate the model at each iteration of the training loop. The remaining 85% is considered the training dataset and is ordered according to the ordering approach (kernel herding or random order). We then fine-tune the chosen pre-trained model iteratively with successively larger portions of the training dataset, increasing by 50 examples per iteration.

Within each iteration of this experimental process, the images are cropped according to the current object detector model. For the BB strategy, we train an object detector each iteration on all current training images, whereas for BB-1 we train the object detector only once on the first iteration of 50 images.

The networks are optimized using gradient descent employing a validation set for early stopping. Once the training dataset for the next iteration has been assembled, we remove a randomly-selected 15% for internal validation. The remain-

[3] http://vision.stanford.edu/aditya86/ImageNetDogs/main.html

[4] http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

ing training items are randomly shuffled and used to fine-tune the pre-trained model using gradient descent.

The initial weights for the pre-trained models are obtained by training on ImageNet, with fine-tuning using the sparse categorical cross-entropy loss function, the ADAM optimizer with an initial learning rate of 0.0001 and a decay of 0.000001, and accuracy as the only metric for early stopping. Fine-tuning is applied for a maximum of 100 epochs of mini-batch stochastic gradient descent with 5 images at a time in each mini-batch used for computing gradients.

After fine-tuning, we evaluate the model against the holdout dataset to estimate the predictive accuracy of the model at that point in the simulated human-in-the-loop training process. Additionally, we evaluate the current model against the 50 selected examples for the next iteration as a measure of how well it performs when pre-labelling those items. This allows us to estimate the number of examples that would need to be corrected if an actual human were involved in the experiments. We can compare the ground-truth labels and the predicted labels in our simulated human-in-the-loop set-up because all the benchmark data used in our experiments is fully labelled.

4 Results of the experiments

As the random and kernel herding methods were generally superior in the previous experiments [5], which evaluated human-in-the-loop training for image classification without bounding boxes, we use them as baselines. Figure 4 shows the overall accuracy of each model on each dataset as the number of iterations progresses, as judged by the holdout test set. All curves tend to the same level of accuracy. Where there are differences, these are typically due to better performance by the BB strategy. Random ordering and kernel herding without object detection, as used in [5], are worse. However, the differences are generally small.

Considering the number of corrections required in each iteration, as in Figure 5, the graphs are generally again quite similar, but the kernel herding set of results shows significant instability. Cumulative corrections, as shown in Figure 6, show more separation in the graphs. This shows that the previous high-performing ordering methods from [5] are improved significantly by the addition of bounding boxes. The best methods are random BB and random BB-1. This pattern is repeated in Figure 7, where the top-left results, indicating high accuracy over the fewest iterations, are generally those same methods. Encouragingly, the two BB-1 methods often perform quite well.

An overview of the total user effort required to fix the labels is shown in Table 2. The table shows that regardless of ordering method, it is always better to use a bounding box strategy than not. The best bounding box strategies are random BB and kernel herding BB. Random BB-1 and kernel herding BB-1 are generally worse than their BB counterparts but perform similarly. These results are promising in the sense that the differences between fully corrected bounding boxes (BB) and single-iteration corrected bounding boxes (BB-1) is not that great. This also suggests that a user interface designed to encourage the user

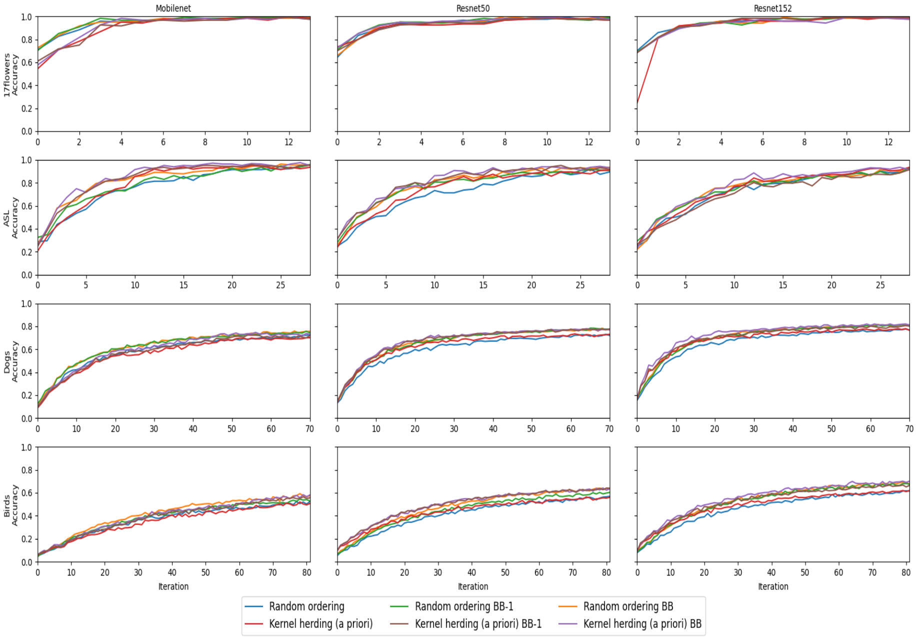


Fig. 4. Holdout accuracy

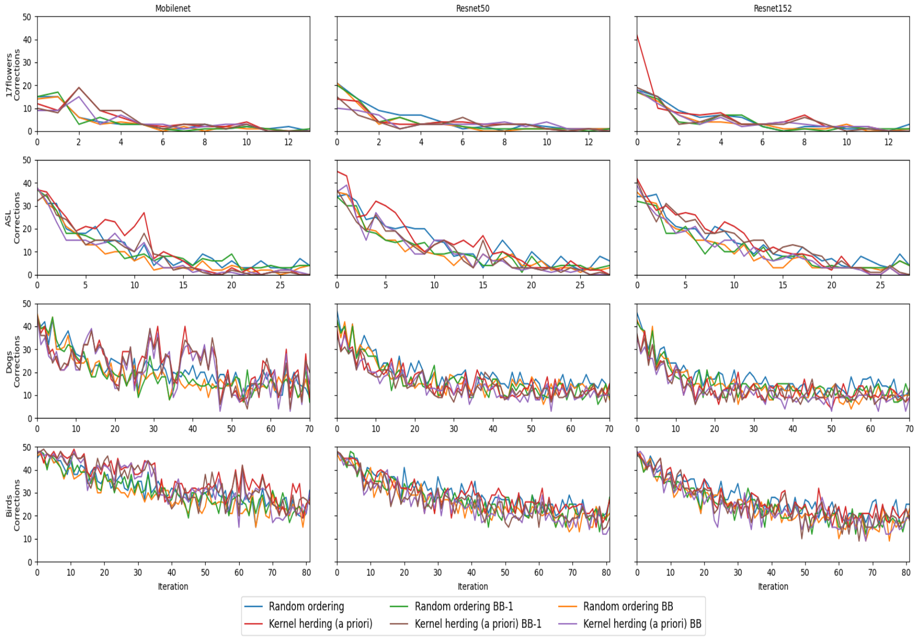


Fig. 5. Corrections needed per iteration

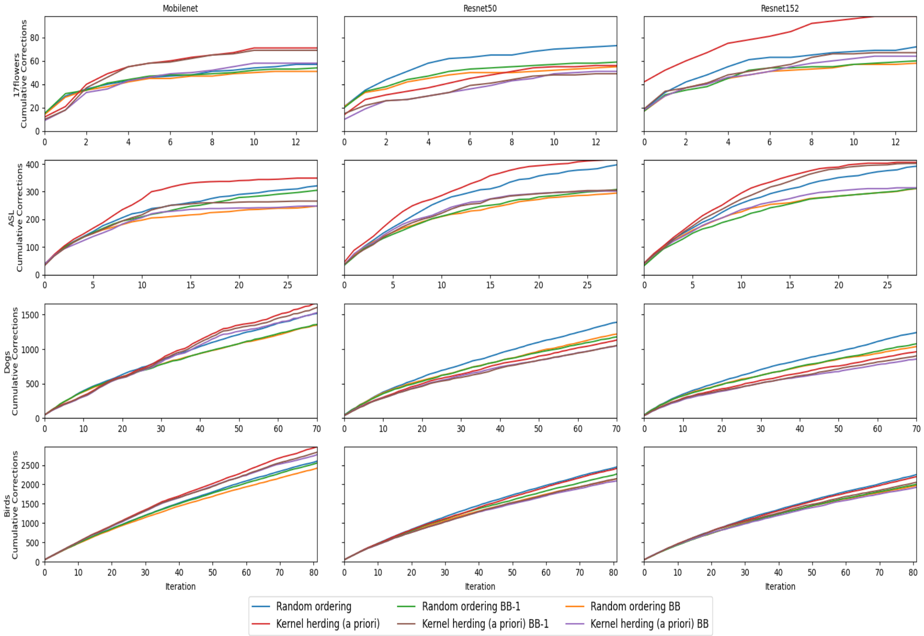


Fig. 6. Cumulative correction counts

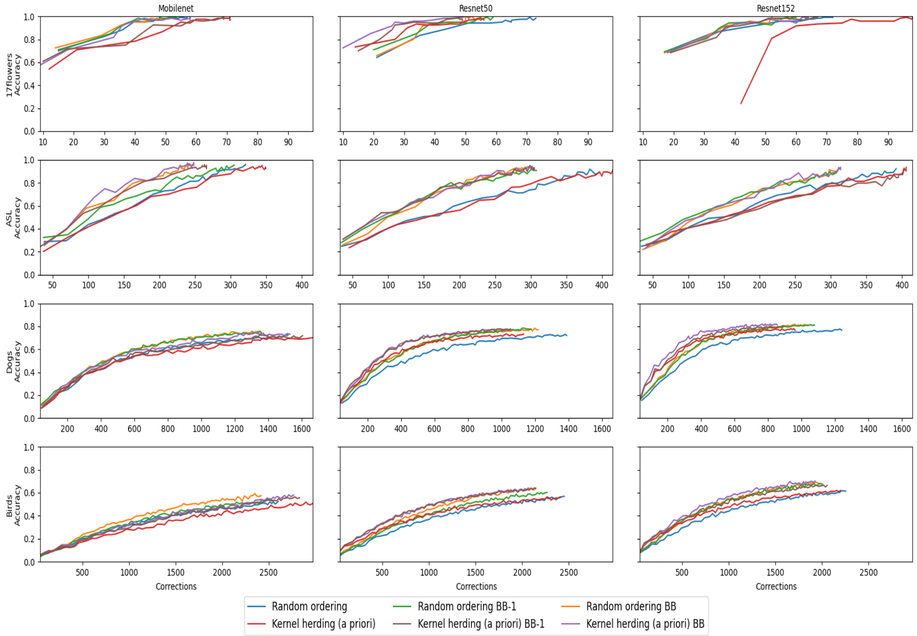


Fig. 7. Accuracy versus number of corrections

Table 2. Total corrections per method and model with and without bounding boxes

Model	Random	Random BB	Random BB-1	KH	KH BB	KH BB-1
17F-mnet	57	51	54	71	58	69
17F-resnet50	73	55	59	56	51	49
17F-resnet152	72	58	60	98	64	67
ASL-mnet	321	248	305	349	248	266
ASL-resnet50	397	295	308	415	302	305
ASL-resnet152	392	311	312	406	314	402
Dogs-mnet	1517	1348	1358	1664	1528	1604
Dogs-resnet50	1390	1219	1178	1133	1048	1054
Dogs-resnet152	1241	1038	1078	963	857	903
Birds-mnet	2603	2419	2554	2972	2768	2832
Birds-resnet50	2453	2147	2269	2415	2090	2148
Birds-resnet152	2255	1974	2003	2201	1926	2055

to only correct the worst cases of bounding box error is likely to lead to good results.

5 Conclusion

A combination of using passive sampling methods alongside class-agnostic object detection for image classification shows that bounding boxes help to reduce the number of class labels a user has to correct in a human-in-the-loop training scenario. Two strategies were adopted representing the maximal and minimal bounding box correction effort that could be made in the loop. While the maximal approach outperformed the minimal approach, the difference in terms of numbers of corrections needed was not significant. This is important because correcting bounding boxes obviously also requires user effort; thus, constructing a human-in-the-loop system for image classification that only requires minimal correction of bounding boxes, namely in the first batch of data, can be recommended as a practical approach.

There are a number of avenues for future work. Our results suggest that effective and simple user interfaces can be constructed using a passive sampling method coupled with a click-on-the-object-of-interest strategy for worst-case bounding box errors. The proposed system also offers the possibility of monitoring and potentially correcting model bias. Investigating different object detection methods in the setting considered in this paper could also be a fruitful undertaking.

References

1. Cassidy, B., Reeves, N.D., Pappachan, J.M., Gillespie, D., O'Shea, C., Rajbhandari, S., Maiya, A.G., Frank, E., Boulton, A.J.M., Armstrong, D.G., Najafi, B., Wu, J., Kochhar, R.S., Yap, M.H.: The DFUC 2020 dataset: Analysis towards diabetic foot ulcer detection. touchREVIEWS in Endocrinology **17**(1), 5–11 (2021)

2. Chen, Y., Welling, M., Smola, A.: Super-samples from kernel herding. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence. pp. 109–116 (2010)
3. Gulli, A., Pal, S.: Deep learning with Keras. Packt Publishing Ltd (2017)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer vision and Pattern Recognition. pp. 770–778 (2016)
5. Holmes, G., Frank, E., Fletcher, D., Sterling, C.: Efficiently correcting machine learning: considering the role of example ordering in human-in-the-loop training of image classification models. In: 27th International Conference on Intelligent User Interfaces. pp. 584–593 (2022)
6. Honeycutt, D., Nourani, M., Ragan, E.: Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions of model accuracy. In: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing. vol. 8, pp. 63–72 (2020)
7. Jaiswal, A., Wu, Y., Natarajan, P., Natarajan, P.: Class-agnostic object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 919–928 (2021)
8. KC, D., Zhang, C.: Improving the trustworthiness of image classification models by utilizing bounding-box annotations. arXiv preprint arXiv:2108.10131 (2021)
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems **28** (2015)
10. Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2012)
11. Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., He, L.: A survey of human-in-the-loop for machine learning. Future Generation Computer Systems **135**, 364–381 (2022)