BiGuide: A Bi-level Data Acquisition Guidance for Object Detection on Mobile Devices

Lin Duan
lin.duan@duke.edu
Duke University

Ying Chen
ying.chen151@duke.edu
Duke University

Zhehan Qu
zhehan.qu@duke.edu
Duke University

Megan McGrath
megan.mcgrath@duke.edu
Duke Lemur Center

Erin Ehmke
erin.ehmke@duke.edu
Duke Lemur Center

Maria Gorlatova
maria.gorlatova@duke.edu
Duke University

ABSTRACT

Object detection (OD) is crucial for numerous emerging visual sensing applications. As OD models trained on unrepresentative data usually yield poor performance, collecting high-quality data in the local environment is recognized to be essential for improving model accuracy. Yet, the question of *how* to collect this data is currently largely overlooked; unsupported data collection tends to produce datasets with a significant proportion of redundant or uninformative data, hindering effective model training. To address this challenge, we design a real-time data importance estimation method and integrate it into *BiGuide*, a bi-level image data acquisition system we create for OD tasks. BiGuide assesses the importance of the captured images in real-time based on *informativeness* and *diversity* estimations and dynamically guides users in collecting useful data via *image-level and object instance-level guidance*. We prototype BiGuide in an edge-based architecture using commodity smartphones as mobile clients, and evaluate its performance via an IRB-approved study with 20 users. Our evaluation demonstrates that OD models trained on the data collected by BiGuide outperform models trained on the data collected by two baseline systems, achieving detection accuracy improvements of up to 33.07% and 14.57%, respectively. Over 85% of the users found BiGuide fast, helpful, and easy to understand and follow.

KEYWORDS

Data acquisition, visual sensing, informativeness and diversity estimation, user guidance, object detection.

1 INTRODUCTION

Accurate object detection (OD) using deep learning plays a crucial role in a wide spectrum of applications, including video analytics [58, 66], autonomous driving [18, 69], and augmented reality (AR) [3, 29]. Despite the existence of OD models pre-trained on large-scale general-purpose datasets such as ImageNet [16] or COCO [28], adapting the model to the task-specific data domain is necessary to achieve high performance in many practical scenarios. Fine-tuning, as well as more advanced domain adaptation methods [25, 52] deployed on task-specific small-scale datasets, have been proven to yield significant performance improvements. However, these methods typically assume the existence of a pre-collected useful and representative target domain dataset. The question of how to obtain such a dataset in the real-world scenario for effective adaptation has unfortunately received only limited attention.

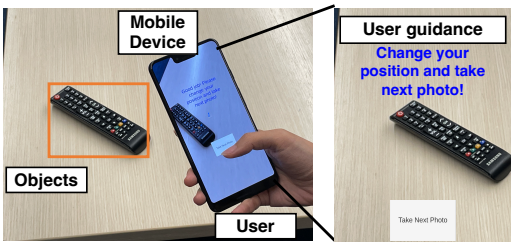

Figure 1: BiGuide in action. The user points the camera toward the objects and observes in-situ guidance.

Unsupported data collection tends to yield datasets with a significant proportion of redundant or uninformative data, which not only wastes human effort but also hinders effective model training. Without appropriate guidance, the performance is inferior even when collecting a large-scale dataset compared to collecting a smaller dataset with proper guidance, as validated in our experiments (§7.4).

To obtain useful data, various *data-importance-aware* acquisition methods have been proposed for environment monitoring [26, 55], spatial mapping [32, 67], and deep learning-based OD tasks [24, 62]. Yet, when deploying these methods to collect target data for building high-performance OD models, two practical challenges arise.

First, some methods [13, 26, 32, 67] employ exhaustive search algorithms to determine optimal sensor locations and angles from preset alternatives to collect useful data. Such approaches necessitate considerable time both to retrain the OD model and to assess the performance changes with every addition of new data to the dataset. This process is impractical during data collection, which demands estimating the importance of the current image and making decisions on its acceptance or rejection in real-time. Moreover, in real-world situations, performance changes are usually not measurable in the absence of labeled data.

Second, some approaches [24, 62] aim to collect diverse data by maximizing the coverage of viewpoints. These methods are constrained by physical obstacles in real-world environments, making it difficult for the collected data to fully cover the desired viewpoints of the objects. Thus, the collected images are sometimes even less valuable than the images collected by users without any constraints, as demonstrated in §7.4.

To move beyond these limitations, we present BiGuide, *the first data acquisition system that instructs users in collecting diverse and informative data for training OD models*, as demonstrated in Figure 1. Unlike previous works that are restricted to predetermined viewpoints, BiGuide sets a new direction by instructing users to collect useful data with flexible and adaptive guidance. It provides both

979-8-3503-6201-5/24/$31.00 ©2024 IEEE
DOI 10.1109/IPSN61024.2024.00012

image- and object instance-level guidance generated based on the estimation of data importance. The *image-level* guidance instructs users to change their camera locations to capture images from different perspectives. The *instance-level* guidance directs users to adjust object poses, enhancing the variety and diversity of the instances in captured data. With this bi-level guidance, users can acquire data that helps the model learn better representation and improves model generalization. Furthermore, BiGuide dynamically adapts the guidance during data collection to ensure positive user experience without sacrificing the usefulness of the collected data.

The core capabilities of BiGuide are made possible by estimating data importance in an online and real-time manner. Some existing deep learning-based methods [6, 22] estimate data importance based on informativeness, such as evaluating the model's confidence in prediction results, while others [8, 14, 20, 60] estimate it based on diversity, by examining distances between samples or by analyzing the distribution of the given dataset to select representative samples that capture its characteristics. Yet, these approaches have limitations when dealing with images captured in succession during data collection. These limitations include the inherent uncertainty of prediction confidence caused by the domain gap, the requirement to access the entire dataset for evaluating representativeness, or the necessity for exhaustive pair-wise data comparison. In our data importance estimation method, we design an adaptive acceptance determination strategy to combat the inherent uncertainty of informativeness estimation and formulate the image- and instance-level diversity scores to assess the data diversity online and in real-time.

We build BiGuide, estimate its latency through system profiling, and conduct an IRB-approved user study with 20 participants to evaluate system effectiveness. To illustrate the versatility of our system, we conduct the user study in different scenarios, including an indoor office environment and an outdoor wildlife exhibit at the Duke Lemur Center. These scenarios ensure a diverse range of data collection situations, considering factors such as locations, lighting conditions, and object variability. We highlight the usefulness of the data collected by BiGuide through both supervised and unsupervised learning approaches. Furthermore, qualitative user feedback indicates that BiGuide is perceived as fast, helpful, and easy to understand and follow. We share the collected datasets and the code via Github[1].

Our key contributions can be summarized as follows:

- We develop BiGuide, a bi-level data acquisition system that generates real-time and in-situ guidance to dynamically instruct users to collect useful data for training accurate OD models.
- We design a data importance estimation method to enable BiGuide to estimate the informativeness and diversity of the successively captured images in an online, real-time manner, to actively guide data collection.
- We implement BiGuide in an edge-based architecture, using commodity smartphones as mobile clients, and evaluate it via a user study with 20 participants in a controlled indoor scenario and a dynamic wildlife scenario with unpredictable conditions. OD models trained using data collected by BiGuide achieve up to 33.07% and 14.57% higher detection accuracy compared to OD models trained with data collected by baseline systems.

[1] https://github.com/BiGuideCollection/BiGuide

Below, we review related work in §2, present the motivation in §3, describe the system overview in §4, introduce the main components of BiGuide in §5 and §6, evaluate BiGuide in §7, and discuss and conclude the paper in §8 and §9.

2 RELATED WORK

Object Detection. OD models can be trained using supervised learning methods [3, 29, 58] with annotations such as object labels and bounding boxes, or unsupervised learning methods [53] without annotations. The standard approach for improving the OD model performance for a specific application or a specific set of conditions is to first pre-train the model on a large-scale general-purpose dataset (such as ImageNet [16] or COCO [28]), and then fine-tune it via a small set of *custom target domain data* collected specifically for the application [40]; this approach is known to lead to dramatic performance improvements over using the pre-trained model without further customization [25, 52]. However, the question of *how to collect* representative target domain data is currently largely overlooked. BiGuide is designed to help in collecting useful data for training high-performance OD models.

Active Data Acquisition. Active data acquisition methods involve user responses [33, 44]. Our work complements existing active methods by developing techniques for data-importance-aware acquisition. Prior works focus on data-importance-aware acquisition for mobile sensors (robots [26, 55], unmanned aerial vehicles [32, 67]) via the so-called *active sensing* in various tasks (spatial mapping [32, 67], environment monitoring [26, 55]). These methods do not involve users and additionally require an exhaustive search for optimal sensor locations and angles from preset alternatives. Some efforts [24, 62] focus on data acquisition guidance for OD tasks, which employ a *coverage-based method* to guide users to capture the object of interest from diverse viewpoints. However, its reliance on preset viewpoints presents an inherent limitation; in real-world applications, these viewpoints are not always accessible due to physical obstacles (e.g., walls, windows). Additionally, not all data collected from different viewpoints contributes to model training. Our evaluations in §7 show that *a coverage-based method not only underperforms compared to BiGuide but also compared to another baseline method that collects data without assistance.*

Active Learning. Active learning [11] maximizes model performance by selecting important data from a pool of collected samples or data streams. In this context, data importance is typically assessed through informativeness and diversity [23]. Informativeness-based approaches assess data importance by evaluating the model's prediction confidence using measures such as entropy, least confidence, and smallest margin [6, 22, 35]. However, relying solely on informativeness can be problematic because domain shift undermines confidence evaluation [27]. Diversity-based approaches [4, 14, 48, 60] select representative samples by examining distances between samples or clustering data and picking samples across clusters. Recent deep active learning developments have emphasized the efficacy of combining informativeness and diversity measures. Weighted-sum methods [9, 61, 63] balance these two objectives with additional hyperparameters or determinantal point processes. Two-stage optimization methods [4, 49] select an informative subset and refine it

Figure 2: Example images from the lemur dataset: (a) jumping lemur, (b) lemur on a cloudy day, (c) lemur behind a cage.

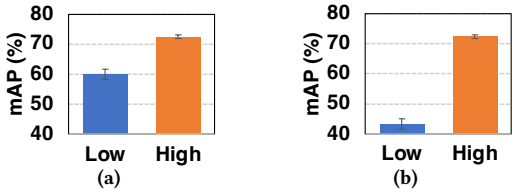

Figure 3: The mAP values of OD models trained on the subsets with low and high (a) informativeness; (b) diversity.

for maximum diversity. These hybrid strategies outperform single-objective methods in various tasks [65], inspiring the design of BiGuide. Yet, these methods either require access to the entire dataset for representativeness evaluation or employ exhaustive pair-wise data comparison, making them *unsuitable for real-time scenarios where new data samples are acquired successively and require prompt evaluation.* We design BiGuide to estimate the informativeness and diversity of images in an online, real-time manner to actively guide data collection.

3 MOTIVATION

Our work is motivated by an ongoing collaborative study with the Duke Lemur Center to deploy a mobile AR app with a robust species (lemurs) detector. The center is home to various lemurs, located in different areas. We are designing our app to detect lemurs in changing environmental conditions, such as after rearrangements of the exhibits, under different weather conditions, in different enclosures, and at different times of the day. When building the detector, we observe data informativeness and diversity to profoundly impact model performance, based on which we further develop our BiGuide system. To showcase our observations, we conduct small-scale experiments using the lemur dataset collected for the aforementioned study.

Lemur Dataset. Lemurs of various species in the Duke Lemur Center are exhibited in rotation under different weather conditions and usually remain active within their spacious enclosures, as depicted in Figure 2. From visitors' viewing points outside these enclosures, we can take images of lemurs with different poses, as lemurs change their positions and activities from time to time. We collected a dataset including four lemur species, not only from the center but also from YouTube videos and two image search platforms offering copyright-free images (Flickr [19] and Wikimedia Commons [56]). The species are black-and-white ruffed lemurs, Coquerel's sifakas, red ruffed lemurs, and ring-tailed lemurs. The dataset includes 499 images of lemurs, with 401 images in the training set and 98 images in the test set. These images encompass lemurs in motion, influenced by different weather conditions. We manually labeled the dataset and shared it via GitHub[1].

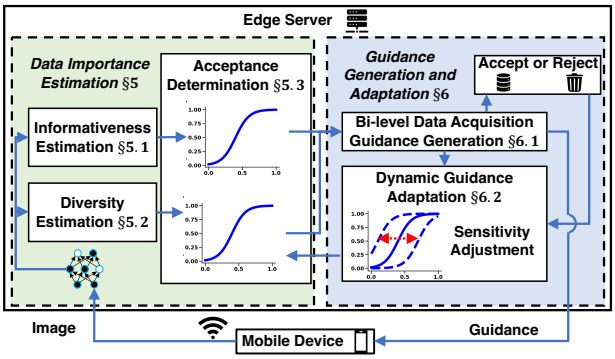

Figure 4: *BiGuide* System architecture.

Experimental Setting. We use Faster-R-CNN [45] with VGG16 backbone pre-trained on the COCO dataset. Performance is measured via mean average precision (*mAP*) with an intersection over union threshold of 0.5 [31].

Impact of Data Informativeness. To estimate data informativeness [6, 22, 35], we measure the model's confidence in its prediction results by the least confidence method [35]. We conduct 3 trials to randomly extract 2 subsets with low (<0.4) and high (≥0.4) informativeness values from the training set, each containing 20 samples. We train the model on these subsets separately. The average model performance impacted by the informativeness values is shown in Figure 3a. When the model is trained on high-informativeness data, its performance shows a *remarkable improvement of 12.55%* compared to the model trained on low-informativeness data, even though there are no obvious visual differences between low- and high-informativeness subsets for the human eyes.

Impact of Data Diversity. To investigate the impact of data diversity [8, 20] on model performance, we conduct 3 trials to manually extract 2 subsets of 20 samples from the training set. One subset is crafted to exhibit high data diversity, characterized by perceptually distinct samples (e.g., with different poses, in varied locations). Conversely, the other subset is designed to exhibit low data diversity (e.g., with similar poses, in comparable locations). We train the model on these subsets separately, and show the average results in Figure 3b. The *mAP* of the model trained on high-diversity data is *28.92% higher* compared to the model trained on low-diversity data. This indicates data with various object appearances and environmental conditions enables training more accurate OD models.

4 SYSTEM OVERVIEW

The overall system architecture of *BiGuide* is shown in Figure 4. BiGuide comprises two major components: data importance estimation and guidance generation and adaptation. They are deployed on the edge server such that no significant computation overhead is introduced on the mobile device. In addition to the server, there is a mobile app running on a mobile device which wirelessly sends images captured by the user to the edge, receives real-time data acquisition guidance from the edge, and presents it to the user, as depicted in Figure 1. The images collected with BiGuide's assistance are used to train OD models, whose performance is evaluated in §7.

Data Importance Estimation. As the server receives an image, BiGuide evaluates its importance through measures of *informativeness* and *diversity* (§5). The informativeness is determined by

the prediction confidence of the OD model (§5.1). To complement the importance assessment, we design the *image-* and *instance-level diversity scores* of the current image to quantify how much it contributes to the diversity of the collected images at both image and instance levels (§5.2). Probability functions are then applied to determine the acceptance or rejection of the image based on the informativeness and the diversity scores (§5.3).

Guidance Generation and Adaptation. Guidance generation and adaptation (§6) consists of two steps: bi-level data acquisition guidance generation (§6.1) and dynamic guidance adaptation (§6.2).

Bi-level data acquisition guidance generation: Based on the acceptance or rejection decision determined above, the server either accepts the captured image and sends feedback to notify the user, or rejects the image and generates image- or instance-level guidance to help the user collect informative and diverse data. Image-level guidance instructs users to change their camera location to get different image backgrounds, while instance-level guidance instructs users to change the object's pose or wait for the object's pose change to get more diverse instances of the objects.

Dynamic guidance adaptation: The recent acceptance or rejection results are recorded to determine the sensitivity of the guidance as the user continues to capture images. When captured images keep getting rejected, the centers of the diversity-based acceptance probability functions are adjusted to increase the acceptance rate. Conversely, if accepted images contribute no new information, the centers are adjusted to decrease the acceptance rate.

5 DATA IMPORTANCE ESTIMATION

In this section, we introduce our data importance estimation approach, which evaluates the informativeness (§5.1) and diversity (§5.2) of the captured images, as depicted in Figure 5. Based on the data importance, we propose an adaptive acceptance determination strategy (§5.3). Note that the applied OD model is pre-trained on the large-scale, pre-existing dataset that contains the same classes as the data being collected. This pre-training equips the model with the capability to extract general features.

5.1 Informativeness Estimation

To evaluate the informativeness of the captured image, we quantify the amount of useful information for model training in the image. Following recent works [6, 22, 35], we measure the image informativeness via the OD model's prediction confidence, which encompasses both class and bounding box prediction confidence of the image. High informativeness indicates that the OD model is uncertain about the prediction results of the image, suggesting that it contains valuable information for the model. Formally, given the ordered set of prediction confidence scores $P = \{p_m\}_{m=1}^{d}$ produced by the OD model, where d represents the number of predicted bounding boxes and p_m is the m-th confidence score in P, we calculate the informativeness I as follows:

$$m^\star = \arg \max_{m \in \{1,\cdots,d\}} p_m, \tag{1}$$

$$I = 1 - p_{m^\star}, \tag{2}$$

where $m^\star$ is the index of the predicted bounding box with the highest confidence score in P. The informativeness estimation process is shown schematically in Figure 5. The core driver of the latency of this component is the inference time to execute an OD model that takes the image captured by the user as input. As the image size

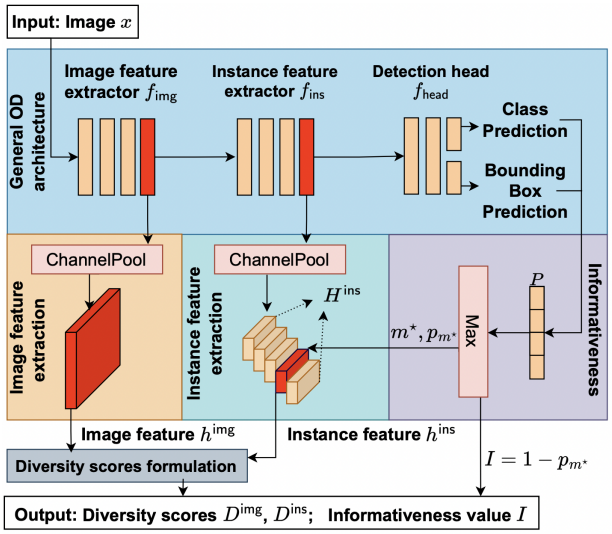

Figure 5: Informativeness and diversity estimations. We obtain the informativeness I based on the prediction confidence and calculate the diversity scores D^{img}, D^{ins} based on the image and instance features extracted by the OD model.

remains constant during the data collection, the time complexity of the informativeness estimation is constant, denoted as $O(1)$.

5.2 Diversity Estimation

To avoid collecting repetitive images, we design a complementary estimation method called diversity estimation. It consists of feature extraction and diversity scores formulation at both image and instance levels.

Bi-level Feature Extraction. To efficiently and accurately evaluate data diversity, we analyze the currently captured image using its low-dimensional embedding in the feature space, which has higher information density than the raw image. In the context of OD, images are typically treated as a combination of background and object instances. Therefore, we analyze bi-level features of the currently captured image, including the *image feature* h^{img} and *instance feature* h^{ins}. Following common practices in the field [52], we extract the *image feature* h^{img} from the last layer of the OD model's backbone, denoted as f_{img}. In addition, to enable more efficient computation while still preserving the crucial information required for accurate diversity estimation, we further reduce the dimension of the extracted feature by channel-wise max pooling:

$$h^{\text{img}} = ChannelPool(f_{\text{img}}(x)), \tag{3}$$

where x is the image fed to the OD model. To obtain the *instance feature*, we first extract a set of instance features H^{ins} using the instance feature extractor f_{ins} immediately preceding the detection head f_{head}, upon which we apply the same channel-wise max pooling to reduce the dimension. This ordered set of *instance features*, paired with the corresponding predicted bounding boxes in the same order, can be expressed as:

$$H^{\text{ins}} = \{h_m^{\text{ins}}\}_{m=1}^{d} = ChannelPool(f_{\text{ins}}(f_{\text{img}}(x))), \tag{4}$$

where h_m^{ins} represents the m-th instance feature in H^{ins}. To avoid analyzing wrongly predicted instances during data collection process in which ground-truth instances are not available, we further *select the instance with the highest confidence* in its predicted bounding box

as the only instance used to form the *instance feature*. Specifically, we select $h^{\text{ins}} = h^{\text{ins}}_{m^\star}$ as the final *instance feature*. The entire bi-level feature extraction process is an inherent part of model inference, entailing no additional computational overhead.

Bi-level Diversity Scores Formulation. With the extracted features, we then formulate the image-level and instance-level diversity scores of x in relation to the collected image set S containing s images. By measuring the distance from the newly added sample to the cluster centers of existing samples and the representative samples, these scores quantify the potential improvement in the diversity at both image and instance levels. Given the similarity of the formulation of image- and instance-level scores, we focus on explaining the details of the *image-level diversity score* as follows.

We first compute a set of prototypes [38, 50] used in the estimation process. We denote the set of extracted image features from S as $H = \{h_i\}_{i=1}^{s}$, where h_i represents the image feature of the i-th image in the image set S. To enable more efficient computation, we then fit the commonly used principal component analysis (PCA) model [1] to H and project vectors in H onto a low-dimensional space to get vectors $V = \{v_i\}_{i=1}^{s}$ with $v_i = PCA(h_i)$. Following the convention in PCA method used for dimensionality reduction [37, 64], we empirically set $|v_i| = 15$ to preserve the variance. After that, we use one of the fastest clustering methods, K-means clustering [57], to divide V into z clusters, where z is determined optimally by the Elbow method [7]. We then obtain the vector sets of different clusters, $\{V_k\}_{k=1}^{z}$, and prototypes (also called cluster centers), $\{c_k\}_{k=1}^{z}$, where V_k and c_k are the vector set and prototype of the k-th cluster.

After obtaining the prototypes of the collected images, we apply the fitted PCA to the image feature h^{img} of x to get the low-dimensional vector v^{img}. To quantify how much the diversity of the image set S can be improved at the image level by adding x to the set, we formulate the image-level diversity score D^{img} of x as:

$$D^{\text{img}} = \frac{1}{2}\left(\min_{i\in\{1,\cdots,s\}} \|v_i - v^{\text{img}}\| + \min_{k\in\{1,\cdots,z\}} \|c_k - v^{\text{img}}\|\right). \tag{5}$$

For the formulation of the *instance-level diversity score* D^{ins}, the only difference with the process above is substituting the image feature with the instance feature. A higher diversity score indicates a larger distance between the feature of the current image or instance and its closest feature, as well as the closest cluster prototype. This signifies that the captured image differs from previous images and can potentially enhance the knowledge learned by the OD model.

The diversity score calculation primarily involves PCA and K-means, with a time complexity of $O(z \cdot s)$, which grows linearly as more data is collected in S. This complexity can be reduced to $O(z)$ by limiting the size of S. Thus, inspired by [39], we adopt the 'first in, first out' (FIFO) principle to limit the size of representative images in S, as recent data is more related to the current environment. With consideration of both efficiency and accuracy for BiGuide, we empirically set the size of S to be 20 images per class.

5.3 Acceptance Determination

To determine whether to accept the image, we formulate acceptance probabilities, which take in the outputs of the informativeness and diversity estimations (i.e., I, D^{img}, and D^{ins}), as illustrated in Figure 4. These acceptance probabilities subsequently serve as inputs for the guidance generation and adaptation component.

Inspired by previous works [17, 21] that use sigmoid functions to adapt reinforcement learning policies to real-world environments, we adopt the sigmoid function in adaptive acceptance determination to respond to changing environments instead of using a fixed threshold. We formulate the sigmoid function-based acceptance probability $P^{\text{info}}_{\text{accept}}$ as a function of the informativeness:

$$P^{\text{info}}_{\text{accept}} = \frac{1}{1 + e^{-b(I-a)}}, \tag{6}$$

where a and b control the center and the width of the sigmoid function, respectively. Similarly, we formulate the image-level (or instance-level) acceptance probability as a function of the image-level (or instance-level) diversity score:

$$P^{\text{img}}_{\text{accept}} = \frac{1}{1 + e^{-b^{\text{img}}(D^{\text{img}} - a^{\text{img}})}}, P^{\text{ins}}_{\text{accept}} = \frac{1}{1 + e^{-b^{\text{ins}}(D^{\text{ins}} - a^{\text{ins}})}}, \tag{7}$$

where a^{img} (or a^{ins}) and b^{img} (or b^{ins}) control the center and the width of the sigmoid function, respectively. For $P^{\text{info}}_{\text{accept}}$ and $P^{\text{img}}_{\text{accept}}$ (or $P^{\text{ins}}_{\text{accept}}$), a lower value of a and a^{img} (or a^{ins}) shifts the sigmoid function to the left, increasing the probability of acceptance. We keep the shape of the sigmoid function in (6) fixed during the data acquisition process, as the pre-trained OD model remains consistent. We also find from experiments that the values of b^{img} and b^{ins}, which determine the width of the sigmoid function in (7), are insensitive to the data collection process. As a result, we only dynamically adjust a^{img} and a^{ins} during the data collection. (6) and (7) are monotonic, smooth, differentiable [30], which avoids undesirable abrupt changes in the acceptance probabilities and ensures that images with high informativeness or diversity are more likely to be accepted. At the same time, for images with highly confident predictions (i.e., with low informativeness scores) or with low diversity scores, $P^{\text{info}}_{\text{accept}}$, $P^{\text{img}}_{\text{accept}}$ and $P^{\text{ins}}_{\text{accept}}$ are larger than 0, which means that these images also have a chance of being accepted. Not rejecting all these images altogether can combat the inherent uncertainty in calculating the prediction confidence (using (2)) of the pre-trained OD model due to the domain shift. This also compensates for the potential unreliability of the diversity estimation (using (5)) resulting from the limited size of S.

Adjusting the acceptance probabilities will change the guidance sent to the user, as we will show in §6.1. Such adjustments influence the user's experience during data collection and the usefulness of the collected data. For instance, the user's experience can be adversely affected if images are rarely accepted. We will provide details on how to dynamically adjust the acceptance probabilities and ensure both the user's comfort and data usefulness in §6.2.

6 GUIDANCE GENERATION AND ADAPTATION

In this section, we describe the bi-level data acquisition guidance (§6.1) and outline the dynamic guidance adaptation (§6.2) during data collection. Details of the guidance generation and adaptation methods are outlined in Algorithm 1.

6.1 Bi-level Data Acquisition Guidance Generation

Based on the acceptance determination, we design the bi-level data acquisition guidance to instruct users' data collection. This guidance, at both image and instance levels, aims to direct users to

Algorithm 1 Guidance generation and adaptation pipeline.

1: **Input:** The experimentally set threshold N for subsequent rejection count;
2: **Initialize:** Initialize the image set S using randomly selected images of the same classes from the pre-existing dataset (e.g., public dataset or self-collected dataset). $a^{\text{img}} \leftarrow 0$, $a^{\text{ins}} \leftarrow 0$. $count^{\text{lev}} \leftarrow 0$;
3: **for** each captured image x **do**
4: Calculate A_{img} and A_{ins} using (1)–(8);
5: // *When both the image-level and instance-level decisions are to accept, accept the image and notify the user to continue the collection*
6: **if** $A^{\text{img}} == 1$ and $A^{\text{ins}} == 1$ **then**
7: Following FIFO principle, update S by adding x;
8: **for** each lev in {img, ins} **do**
9: // *Reset the subsequent rejection count to* 0 *once an image is accepted*
10: $count^{\text{lev}} \leftarrow 0$;
11: Update clusters in S;
12: **if** the number of vectors in each cluster remains the same before and after the update **then**
13: $a^{\text{lev}} \leftarrow a^{\text{lev}} + 0.05$;
14: The server notifies the user to continue collecting images;
15: **else**
16: **for** each lev in {img, ins} **do**
17: **if** $A_{\text{lev}} == 0$ **then**
18: $count^{\text{lev}} \leftarrow count^{\text{lev}} + 1$;
19: **if** $count^{\text{lev}} == N$ **then**
20: $a^{\text{lev}} \leftarrow a^{\text{lev}} - 0.05$;
21: $count^{\text{lev}} \leftarrow count^{\text{lev}} - 1$;
22: **else**
23: $count^{\text{lev}} \leftarrow 0$;
24: The server sends the bi-level guidance based on the rule in Table 2;
25: **if** the assigned number of images has been collected **then**
26: End the data collection process;

Table 1: Bi-level data acquisition guidance generation based on the image- and instance-level determinations.

A^{img} \ A^{ins}	1	0
1	Accept; notify the user to continue	Reject; send instance-level guidance
0	Reject; send image-level guidance	Reject; send image- or instance-level guidance

collect both informative and diverse data. Specifically, to generate the *image-level guidance*, we first make an image-level decision A^{img} about whether to accept the captured image. With the acceptance probabilities $P^{\text{info}}_{\text{accept}}$ and $P^{\text{img}}_{\text{accept}}$, we determine A^{img} as:

$$A^{\text{img}} = \begin{cases} 1, & \text{if } rand(0,1) \leq P^{\text{img}}_{\text{accept}} \cdot P^{\text{info}}_{\text{accept}} \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $rand(0,1)$ generates a random number uniformly distributed on $[0,1]$. When A^{img} is 1, the currently captured image is accepted due to its higher chance of being informative and diverse at the image level. Otherwise, when A^{img} is 0, the image is rejected. Then, following the guideline in Table 1, we send image-level guidance. The image-level guidance instructs users to capture images from different perspectives, thereby increasing the image-level diversity of the collected data. To achieve this, we conducted a small-scale user study to observe users' behaviors during data collection and, based on our findings, formulated the specific image-level instructions to be randomly sent to users, as outlined in Table 2. It motivates users to explore different locations and adjust camera angles to capture images from distinct perspectives.

To generate the *instance-level guidance*, we follow the same procedure to make the instance-level decision A^{ins} using the acceptance probabilities $P^{\text{info}}_{\text{accept}}$ and $P^{\text{ins}}_{\text{accept}}$. Then, as shown in Table 2, when A^{ins} is 0, the instance-level guidance is sent to users. It instructs

Table 2: Details of image-level and instance-level data acquisition guidance generated by BiGuide.

	Image-level data acquisition guidance		
G1	Change your position.	G3	Raise your phone.
G2	Tilt your phone.	G4	Lower your phone.
	Instance-level data acquisition guidance		
G5	Change the object's pose or wait for the object's pose change.		

them to capture images with diverse object poses and appearances to enhance the variation of the captured data. If users do not adhere to the guidance when capturing a new image, their captured data is at risk of being rejected due to a low diversity score. In this case, BiGuide sends updated data acquisition guidance based on the new image for users to follow until a captured image gets accepted.

6.2 Dynamic Guidance Adaptation

As discovered in the small-scale user study, a high image acceptance rate allowed the users to collect data quickly and with ease, but the users tended to not move around much, resulting in repetitive data. In contrast, a low acceptance rate forced the users to explore more viewpoints, leading to more diverse data, but also resulting in user frustration and dissatisfaction. We thus enhance BiGuide with the *dynamic guidance adaptation* method described below.

Algorithm 1 summarizes the adaptation of the acceptance probabilities in response to different situations. Let $\text{lev} \in \{\text{img}, \text{ins}\}$ signify the image- or instance-level. In the first situation, the centers a^{img} and a^{ins} of acceptance probabilities $P^{\text{img}}_{\text{accept}}$ and $P^{\text{ins}}_{\text{accept}}$ are dynamically adjusted based on the count of subsequently rejected images at each level (lines 17-23). We denote the count of subsequently rejected images as $count^{\text{lev}}$. When $count^{\text{lev}} \geq N$, with N being the threshold for the subsequent rejection count, the center a^{lev} of $P^{\text{lev}}_{\text{accept}}$ is slightly decreased by 0.05 to increase the image acceptance rate. We experimentally set N to 3 to ensure good user experience and the usefulness of the collected data. During the initial explorations, we found that this gradual increase in the acceptance rate was perceptible to users, effectively reducing fatigue while maintaining an acceptable standard of data quality. In the second situation, if accepted images do not contribute new information to the collected image set S, a^{lev} is slightly increased by 0.05 to decrease the image acceptance rate (lines 10-13). To determine whether an accepted image brings new information, we calculate the difference in the number of vectors within each cluster before and after the cluster update. This difference indicates the variation in clusters of data features after adding the newly captured image. This adaptive approach ensures a positive user experience without sacrificing the data importance of the collected images.

7 SYSTEM EVALUATION

7.1 Experimental Setup

System Implementation. We implement BiGuide in an edge-based architecture using three commodity smartphones as the mobile clients: (1) Google Pixel 3 XL, (2) Google Pixel 7, and (3) Samsung Galaxy Note10+. We design a mobile app on smartphones running Android 11 using Unity 2020.3.14f and ARCore 4.1.7. Data

importance estimation (§5) and guidance generation and adaptation (§6) are executed on the edge server with an Intel i7 CPU, an NVIDIA 3080 Ti GPU, and 64GB DDR5-4800 RAM. For data importance estimation, we employ YOLOv5 [68] for fast OD model inference, which is *independent from OD models trained post data collection*. Communication between the server and smartphones occurs over one-hop 5 GHz WiFi (802.11n), with images resized to $3\times1480\times720$ and JPEG compressed to reduce latency. We select the hyperparameters outlined in §5 and §6 via a small-scale user study based on user perception of the quality of their experience. These parameters are kept the same in all experiments, demonstrating their consistency and insensitivity across various scenarios.

Baselines and Variant. We benchmark the performance of BiGuide against two baselines and one BiGuide variant. All these systems use the same mobile app but differ in the guidance generation process on the server. The baselines are as follows:

(1) Coverage-based system (*CovGuide*). Inspired by [24], CovGuide employs coverage-based guidance to instruct users to capture images from a set of pre-defined viewpoints that cover the entire object as comprehensively as possible.

(2) Free Guidance system (*FreGuide*). The FreGuide system can be viewed as a variant of BiGuide, without modules of data importance estimation and guidance generation and adaptation. Instead, during the pre-study briefings, FreGuide's users are told to move frequently and collect data as diverse as possible. FreGuide transfers the freely captured images from the mobile phone to the server and generates guidance that notifies users to continue capturing images.

We also investigate the importance of guiding users with both image- and instance-level guidance by comparing BiGuide with *ImGuide*, which provides only image-level guidance generated based on informativeness and image-level diversity estimations.

To fairly compare the importance of the data collected by all systems, we instruct users to capture an equal number of images for each class and ensure that objects of interest are clearly in view.

Evaluation Metrics. We evaluate BiGuide on a collection of quantitative and qualitative metrics.

Data usefulness: To assess the usefulness of the collected data, we evaluate the accuracy of OD models trained using both supervised and unsupervised learning methods. For the supervised learning methods, we manually label all the data collected by users (4400 images in total) and train two different OD models, YOLOv5 and Faster-R-CNN, which are independent from the OD model used in BiGuide's data importance estimation. For the unsupervised learning methods, we adopt the SOTA method - CutLER [53]. To evaluate the performance of these models trained on the collected data, we pre-collected 110 images for each class under varying lighting and weather conditions to ensure fairness in the evaluation results. In total, we amassed 770 images in the indoor test set and 330 images in the wildlife test set, as presented in §7.2. With the diverse test set, we measure the system performance via *mAP* as introduced in §3. In all experiments, we apply the commonly used data augmentation methods [34] and use SGD with a learning rate of 0.001 as the optimizer to train the models for 50 epochs.

Other metrics: To examine the system efficiency, we measure the communication and computation latency (in *ms*) of BiGuide. To measure users' movements during data collection, we use the motion sensors embedded in mobile devices and record the data

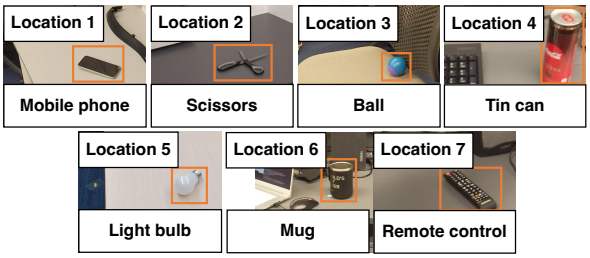

Figure 6: Example images of 7 objects positioned in 7 locations in the indoor scenario. These objects were placed in a controlled environment.

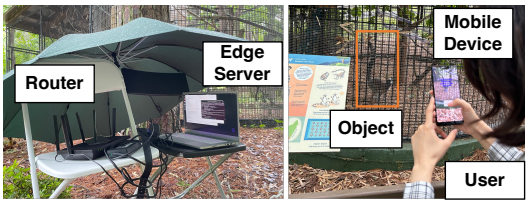

Figure 7: System setup in the Duke Lemur Center.

through our mobile app. With the recorded data, we analyze the average change of Euler angles per second ($degree/s$) to assess the user's movement. A larger change in Euler angles per second indicates more frequent and pronounced movement. We also assess user engagement, preferences, and users' satisfaction via a Qualtrics-based [42] post-experience questionnaire, as detailed in §7.6.

7.2 User Study Setup

User Study Scenarios. To evaluate the performance and the generalization ability of BiGuide, we conducted an IRB-approved user study encompassing two distinct scenarios. We chose the two scenarios to represent potential real-world use cases of BiGuide, particularly catering to individuals or small business stakeholders who, due to limited budgets or lack of experience in extensive data collection and model training, are in search of a lightweight solution for their specific object detection needs. These applications are often built on personalized data, which can significantly differ from the public datasets used to pre-train OD models. Therefore, it's crucial to collect data that is both informative and diverse, enabling the effective fine-tuning of an OD model with reduced effort.

Indoor scenario: Users were guided on collecting data in an indoor environment, which is commonly encountered in OD tasks (e.g., interaction with objects in AR applications [10]). Inspired by the commonly used public indoor-focused CORe50 dataset [15], we set up the indoor scenario in a typical office environment. We included seven object classes that are also present in CORe50, namely: mobile phone, scissors, ball, tin can, light bulb, mug, and remote control. These objects were placed in seven distinct locations within a controlled environment (see Figure 6). Users moved around different locations to collect images of the objects.

Wildlife exhibits scenario: To evaluate our system in a more challenging environment, we set it up in the Duke Lemur Center (see §3), as depicted in Figure 7. This scenario involves outdoor scenes with dynamic objects, specifically lemurs. Users were tasked with capturing images of three lemur species in the center: blue-eyed black

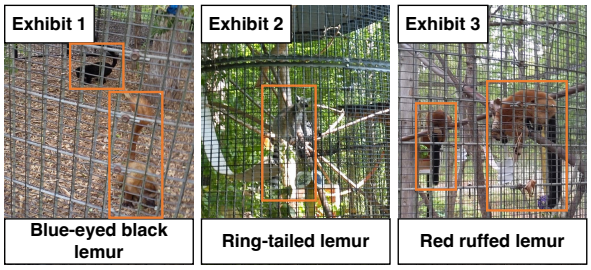

Figure 8: Example images of 3 lemur species enclosed in 3 distinct exhibits in the wildlife exhibits scenario. Images obtained in this scenario are more complex due to lemurs' varying poses and sizes, as well as diverse backgrounds.

Table 3: Communication latency when using Google Pixel 3 XL, Google Pixel 7, and Samsung Galaxy Note10+.

	Image encoding (*ms*)	Info. transm. (*ms*)
Google Pixel 3 XL	77	87
Google Pixel 7	49	
Samsung Galaxy Note10+	55	

lemur, ring-tailed lemur, and red ruffed lemur, as depicted in Figure 8. Different lemur species were housed in distinct enclosures, requiring users to move between these separate areas. Users' visits were scheduled at different times on seven days, aligning with the center's general tour schedule. This led to users encountering different weather conditions, including sunny and rainy days. On warm, sunny days, the lemurs are more active, engaging in activities like climbing and exploring; on cold, rainy days, the lemurs tend to gather and rest inside their cages. Compared to the images collected in the indoor scenario, the wildlife images present greater complexity and detection challenges due to the lemurs' varied poses and sizes, occlusion from cages, and unstable lighting conditions [36].

Study Protocol. For the evaluation of BiGuide, we recruited 20 participants from the Duke community, comprising 12 males and 8 females, ranging in age from 19 to 35 years; 10 participants were assigned to the indoor object scenario, and the remaining 10 to the wildlife exhibits scenario. During the data collection process, each participant collected 20 images for each object in their assigned scenario, using BiGuide and one or more of the alternative systems described in §7.1 above. In the indoor scenario, where the environment was controlled, the viewpoints for CovGuide were marked on the floor, positioned 1 meter away from the object, and spaced at regular 15-degree intervals to ensure comprehensive coverage. To fully cover the viewpoints for CovGuide in the wildlife exhibits scenario, where users' activities are restricted to certain walkways at a distance from the lemurs, users were instructed to move one step forward in these designated areas after capturing each image. We prepared a set of system usage instructions, which can be found on GitHub[1]. Each participant began by reviewing these instructions and was told to collect images with as much diversity as possible. For BiGuide, we explicitly instructed users to follow the guidance displayed on the phone screen. Subsequently, they embarked on the data collection process using the data collection systems. The participant moved around the scene with the commodity smartphone we provided (Google Pixel 3 XL, selected from the range of mobile phone models we tested to best represent a less expensive device), taking images of various objects from different viewpoints. We measured that each participant spent around 15 minutes using each system. For example, in the indoor scenario, to collect 140 images, it required 5 – 11 minutes for CovGuide, 4 – 14 minutes for FreGuide, and 8 – 15 minutes for BiGuide. This results in a total data collection time of over 10 hours.

7.3 System Profiling

We examine the communication and computation latency of BiGuide and baselines.

Communication Latency. Communication in our system, encompassing on-device image encoding and information transmission via WiFi, involves sending images from the device to the server and guidance back to the device. Our communication latency measurement begins when the user presses the screen button and ends when the device receives the guidance, excluding the server's guidance computation time. We measure communication latency over 10 trials with Google Pixel 3 XL, Google Pixel 7, and Samsung Galaxy Note10+ and show the results in Table 3. The image encoding latency when using Google Pixel 3 XL is 77 *ms*, which is longer compared to Google Pixel 7 (with a latency of 49 *ms*) and Samsung Galaxy Note10+ (with a latency of 55 *ms*). We therefore use Google Pixel 3 XL during the user study to demonstrate that BiGuide can be used on lower-end mobile devices without compromising its performance and user experience. The communication latency can be further reduced by decreasing the image size, or adopting wireless networks with higher channel throughput.

Computation Latency. CovGuide and FreGuide have no computational overhead, as no computation happens on the server. The computation latency of BiGuide consists of data importance estimation, guidance generation and adaptation. The average latency over 10 trials is 50 ms when using YOLOv5 in the data importance estimation. When integrating Faster-R-CNN with BiGuide, latency increases by 115 *ms* compared to YOLOv5. Thus, we use YOLOv5 in the user study to provide faster feedback. This computation latency could be further decreased by using lighter OD models, more powerful edge servers, or reduced image size.

End-to-end Latency. The end-to-end per-image-capture latency experienced by BiGuide's users during our user study is 214 ms on average, with a median of 172 ms. Such latency is perceived to be "quick" by more than 85% of the users as reported in §7.6. This can be attributed to the typical user behavior, where users spent about 2328 ms ± 661 ms adjusting their positions between consecutive image captures, allowing BiGuide's guidance to appear promptly before users could determine their next step. Moreover, the nature of our photo-taking task reduces latency impact. It aligns closely with low attention tapping tasks, such as tapping a location and expecting simple feedback, where a latency of 300 ms is acceptable [46]. It also mirrors interactions where users initiate an action (like a button press or voice command) and await visual feedback; in such cases, a latency of up to 200 ms often goes unnoticed [43, 47, 51].

7.4 Identification of the Best Baseline with a Small-scale User Study

We conducted a small-scale user study to identify the best baseline for comparison with BiGuide. In this study, we recruited one user in

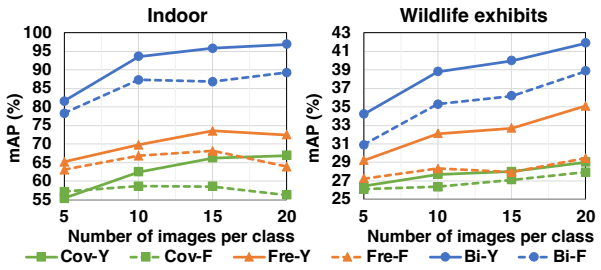

Figure 9: The *mAP* of the YOLOv5 (-Y) and Faster-R-CNN (-F) trained using supervised learning with varying numbers of images collected using CovGuide (Cov), BiGuide (Bi) and FreGuide (Fre). BiGuide outperforms CovGuide by 4.47% ∼ 33.07%**. FreGuide also outperforms CovGuide by** 0.83% ∼ 9.92%**.**

the indoor scenario and one user in the wildlife exhibits scenario to collect data using BiGuide and both CovGuide and FreGuide baselines. The *mAP* of the OD models trained using supervised learning, with varying numbers of images per class collected using different systems, is shown in Figure 9. *BiGuide outperforms CovGuide across all OD models and all training data sizes, with an improvement ranging from* 4.47% *to* 33.07%. Remarkably, for all situations, the OD models trained with images collected by BiGuide, even with as few as 5 images per class, consistently surpass the performance of models trained with a larger set of 20 images per class collected by CovGuide. The improvement ranges from 2.94% to 22.11%. These findings indicate that *BiGuide requires less collection effort while achieving higher performance compared to CovGuide.* Surprisingly, FreGuide also outperforms CovGuide with improvements ranging from 0.83% to 9.92%. We believe the spatial constraints imposed by both scenarios were the primary reason for the poor performance of CovGuide. Taken from a set of fixed viewpoints, the images collected under CovGuide lack diversity and variability, especially in settings where the scene is complex and objects of interest are dynamic, such as at the Duke Lemur Center. Moreover, not all data collected from different viewpoints contribute to model training. As a result, OD models trained with these data struggle to generalize well to different viewpoints and variations in the real world, where users detect objects with mobile devices. Therefore, we identify that FreGuide serves as the most suitable baseline for comparing the performance of BiGuide. We also observe that while FreGuide and CovGuide show no improvement or even degrade with increased data collection, BiGuide consistently improves. This distinction lies in BiGuide's encouragement of diverse data collection, contrasting with the tendency of FreGuide and CovGuide to limit diversity.[2]

7.5 Data Usefulness in Supervised and Unsupervised OD Model Training

To comprehensively validate the usefulness of the collected data, we conduct a user study with 20 users to collect data to train OD models using both supervised and unsupervised approaches.

Data Usefulness in Supervised OD Model Training. The *mAP* of the OD models trained using supervised learning, with varying numbers of images per class collected by FreGuide and BiGuide,

[2]The data distribution comparison can be found in our GitHub.[1]

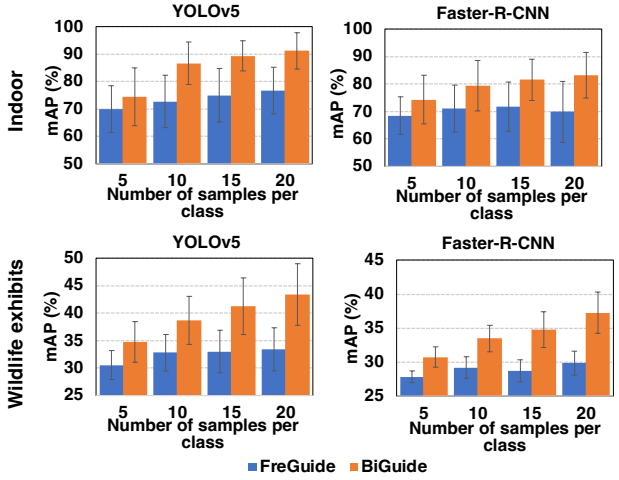

Figure 10: The *mAP* of the models trained using supervised learning with varying numbers of images per class collected by FreGuide and BiGuide.

Table 4: The average *mAP* gap between all OD models trained with data collected by FreGuide and BiGuide under non-expert group and expert group.

Scenarios	Non-experts	Experts
Indoor	+10.74	+11.18
Wildlife exhibits	+9.28	+6.54

is shown in Figure 10. We observe that BiGuide consistently surpasses FreGuide by 4.53% ∼ 14.57% in the indoor scenario and 2.86% ∼ 10.00% in the wildlife exhibits scenario. Notably, as the number of training samples per class increases from 5 to 20, the performance gap between BiGuide and FreGuide becomes more pronounced. For instance, when evaluating the data using YOLOv5 and Faster-R-CNN in the indoor scenario, the performance gap increases from 4.53% to 14.57% and from 5.80% to 13.23%, correspondingly. Similarly, in the wildlife exhibits scenario, the gap widens from 4.21% to 10.00% and from 2.86% to 7.39%. This suggests that users tend to collect redundant data without guidance, whereas BiGuide ensures data diversity and usefulness by employing dynamic data acquisition guidance during the data collection process.

We further analyze the above results with respect to the level of user expertise and user movement.

Analysis based on user expertise: Users were identified to be *experts*, who self-identified in the pre-experiment survey as having a strong machine learning background and being aware of the need to collect diverse data to train models, and *non-experts*. 3 out of 9 experts were assigned to the indoor scenario and 6 experts were assigned to the wildlife exhibits scenario. We expected that the usefulness of data collected by non-experts would be boosted noticeably when using BiGuide, while those collected by experts might only show a smaller improvement. To analyze the difference in the usefulness of collected data between experts and non-experts, we average the supervised learning results across both YOLOv5 and Faster-R-CNN models, as well as all training data sizes in two user study scenarios. The analysis results are presented in Table 4. Surprisingly, BiGuide consistently outperforms FreGuide by 6.54% ∼ 11.18% in expert groups and 9.28% ∼ 10.74% in non-expert

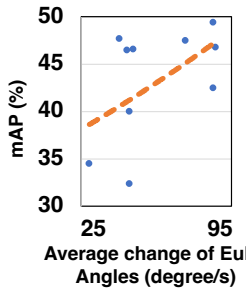

Figure 11: Model performance v.s. average change of Euler angles per second for each user in the wildlife exhibits scenario when using BiGuide.

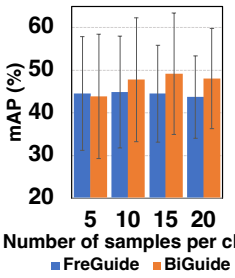

Figure 12: *mAP* of models trained using unsupervised learning with varying numbers of data collected per class by FreGuide and BiGuide in the indoor scenario.

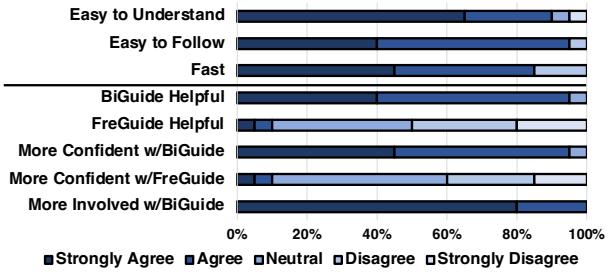

Figure 13: User study response. Users valued the quick response and the ease of understanding and following the guidance, with 85%, 90% and 95% positivity rates, respectively.

groups. It suggests that *even users who have a deep appreciation for the need to collect diverse data still capture redundant data when collecting images without guidance*, whereas BiGuide significantly enhances the data usefulness with in-situ guidance.

Analysis based on user movement: We analyzed user movement in the wildlife exhibits scenario, as movements here are more pronounced compared to the indoor scenario, largely due to the large area of the Duke Lemur Center. We found that users exhibited more significant movements when using BiGuide compared to FreGuide, with average Euler angle changes of 93 $degree/s$ and 42 $degree/s$, respectively. This finding suggests that BiGuide effectively encourages users to actively engage in data collection, resulting in the acquisition of more useful data. Next, we examined the extent of the movement among different users who used BiGuide, as shown in Figure 11. By fitting an exponential trendline, we observe that *users who displayed frequent movement during the user study achieved higher mAP in the models trained on the data collected by BiGuide.* This further indicates that BiGuide provides helpful guidance to instruct users to move more frequently. Furthermore, we notice that the best-performing model (49.4 *mAP*) was trained with data from an active user (91 $degree/s$), while the worst-performing model (32.4 *mAP*) used data from an inactive user (49 $degree/s$). These findings reinforce the importance of user activity in data collection and highlight the benefits of active participation facilitated by BiGuide.

Data Usefulness in Unsupervised OD Model Training. As unsupervised learning often achieves lower performance than supervised learning [12], data usefulness evaluation using unsupervised learning is completed as a secondary use case in the indoor scenario alone, given that the challenging nature of the wildlife data (e.g., moving objects, varying weather conditions, and complex backgrounds) demands more training data to build accurate models.

Figure 12 shows the *mAP* of the OD models trained using unsupervised learning, with varying numbers of images per class collected by FreGuide and BiGuide in the indoor scenario. We observe the best unsupervised result to be 41.98% lower than the best supervised result, which aligns with the performance drop (around 40%) observed in studies comparing supervised and unsupervised methods on the COCO dataset [28, 54]. We also notice that data collected by BiGuide contributes more to the unsupervised training. Specifically, BiGuide outperforms FreGuide by 2.92% ∼ 4.66% as users collect more than 5 images per class, and underperforms FreGuide by 0.71% when collecting 5 images per class. It proves that data collected by BiGuide is helpful for unsupervised learning, even when the data size is small (up to 140 images in the indoor scenario).

7.6 User Study Survey Analysis

Survey Questions. Participants filled pre- and post-experiment online surveys created with Qualtrics software [42]. These surveys are available on GitHub[1]. In the pre-experiment survey, to understand users' expertise, we collected demographic information asking questions about prior experience with object detection and data collection. We assembled a set of questions in different categories for the post-experiment survey to gather feedback. For the category of data acquisition guidance, we asked the participants if the designed guidance was easy to understand and follow and if the guidance generation was fast. For the category of system experience, we asked the participants if the system was helpful and if they felt more confident and more involved when using the system. All questions in these categories were answered on a five-point Likert scale. Finally, participants chose their preferred system and provided open-ended feedback about their overall experience.

Survey Responses. The post-experiment survey responses are summarized in Figure 13. We define *positivity rate* as the percentage of users' responses in the 'strongly agree' and 'agree' categories. The users' free-text responses are quoted with the participant number, *P*.

System experience: The users largely agreed that BiGuide was helpful in collecting diverse data, making them feel more confident and engaged. **100%** of participants expressed a higher level of engagement when using BiGuide compared to FreGuide. **95%** of participants found BiGuide to be helpful in collecting useful data, and reported feeling more confident about the usefulness of the collected data. In contrast, only 10% of participants were confident in the usefulness of the data collected without guidance.

Data acquisition guidance: During the user study, the majority of participants appreciated the ease of understanding and following the provided guidance, as well as the system's fast guidance generation process (90%, 95% and 85% positivity rates, correspondingly). This highlights BiGuide's potential for high-quality data collection.

Table 5: Ablation study results under two evaluation scenarios. BiGuide surpasses ImGuide by 5.65% ∼ 22.54%**.**

	Indoor scenario		Wildlife exhibits scenario	
	YOLOv5	Faster-R-CNN	YOLOv5	Faster-R-CNN
CovGuide	66.88	56.23	29.00	27.93
FreGuide	72.51	63.84	35.11	28.43
ImGuide	74.35	68.81	36.28	32.47
BiGuide	**96.89**	**89.30**	**41.93**	**38.87**

One participant expressed dissatisfaction with the speed and comprehensibility of the guidance, specifically mentioning issues with the small and hard-to-see buttons and guidance text in the interface (P1). These user interface issues stem from the limited-size phone screen, where the guidance information is overlaid on the captured image. Two other participants (P2 and P10) disagreed that the guidance was fast because they felt uncomfortable pausing to see feedback while taking pictures. The user experience can be further enhanced by optimizing the app's user interface.

System preference: A majority (60%) of users favored BiGuide, 35% found merit in both systems, and only 5% preferred FreGuide. Participants who liked BiGuide felt that BiGuide is 'really promising' (P6, P7, P17) and stated that BiGuide 'made me more creative in changing the object's pose' (P16, P17, P18), and 'forced me to move' (P1, P2, P8, P11). Users who liked FreGuide felt that FreGuide is 'convenient' (P1, P2, P10, P15), since they 'only need to press the button' (P1, P11). These insights highlight BiGuide's potential as a tool that not only facilitates practical data collection but also stimulates user engagement and creativity.

7.7 Ablation Study

Comparison with Single-level Guidance. To validate the necessity of using bi-level data acquisition guidance instead of single-level guidance during data collection, we conduct an ablation study by examining BiGuide with only image-level guidance, referred to as ImGuide. The same user study process was conducted by one system designer in both scenarios. The *mAP* of the OD models trained using supervised learning, with 20 images per class collected by CovGuide, FreGuide, ImGuide, and BiGuide, is shown in Table 5. We observe that ImGuide outperforms FreGuide by 1.17% ∼ 4.97%, and BiGuide surpasses ImGuide by 5.65% ∼ 22.54%. These findings demonstrate that both image-level and instance-level guidance are helpful in data collection process.

The Impact of Data Augmentation. Data augmentation can also be considered as a procedure to increase data diversity. To underscore the advantages of using BiGuide for complementing the conventional and SOTA data augmentation techniques [34], such as affine transformation, flipping, HSV augmentation, mixup, AutoAugment, and Copy-Paste, we conduct an empirical evaluation. This entails comparing the performance of YOLOv5 models trained with and without data augmentation. We use data collected by FreGuide and BiGuide as detailed in §7.4. The *mAP* of the OD models, trained using supervised learning with 20 images per class, both with and without data augmentation, is shown in Figure 14. We observe that data augmentation enhances model performance by 3.47% ∼ 22.50%, demonstrating its pivotal role in model training. Meanwhile, with data augmentation, models trained on data

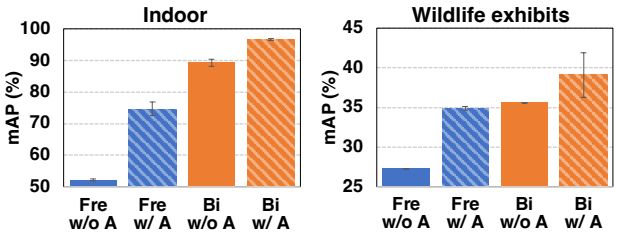

Figure 14: Results with data augmentation (w/ A) and without data augmentation (w/o A).

collected by BiGuide outperform models trained on data collected by FreGuide by 4.22% ∼ 21.94%. This occurs because, while data augmentation can enrich the data distribution [59], it does not offer additional insights into the local environment. In contrast, data collected by BiGuide encompasses more local information, enhancing model performance and serving as a valuable complement to data augmentation techniques.

8 DISCUSSION

Reliability of the Informativeness Estimation. Given that the OD model remains unchanged during the data collection process, there is a potential for unreliability in informativeness estimation. To counter this, we have implemented three approaches to ensure the reliability of this estimation. Firstly, during data collection, users were required to collect a predetermined number of images for each category, eliminating class bias. Secondly, to counteract data bias resulting from repetitive images with high informativeness, we integrated a diversity metric. Thirdly, due to the unacceptable latency incurred by updating the OD model during data collection (4.6 min to update the Faster R-CNN model on NVIDIA TITAN RTX GPU, as found in our previous study), we abstained from runtime OD model updates and introduced randomness into the determination of image acceptance in §5.3, inspired by the error injection method [5]. This reduces reliance on pre-trained OD models and mitigates data bias. For scenarios where users cannot control the number of images per category, can accommodate model update delays, or possess more powerful edge servers, BiGuide can be readily extended to employ SOTA bias-aware informativeness metrics [2, 41] or incorporate advanced model update techniques from active learning research [11, 14, 60].

Refinement of Movement Instructions. In our work so far we ensure the effectiveness of BiGuide by rejecting redundant images captured when participants do not move in accordance with the guidance. However, we do not prescribe specific movements for the user and do not explicitly monitor her adherence to our guidance. In our future work, we will enhance BiGuide's data acquisition guidance by incorporating more precise instructions, such as the specific direction of user motion or the specific degree by which the user needs to adjust her phone. Inspired by active sensing techniques [55, 67], we plan to achieve more explicit guidance by analyzing the impacts of different user actions on the data importance, and identifying a user action that maximizes the data importance to guide user behavior. We will begin by designing a discrete action space that lists possible user actions. Following this, we will formulate a tractable utility function that quantifies data importance

as a function of these actions, and solve the utility maximization problem to obtain the optimal action.

9 CONCLUSION

We develop a novel data acquisition system, *BiGuide*, to instruct users in gathering useful data in the local environment for OD model training. To achieve this, we propose a data importance estimation method to assess the value of the captured image in real-time, a bi-level data acquisition guidance to involve users in data collection, and a dynamic guidance adaptation to ensure a positive user experience without sacrificing the usefulness of the collected data. We implement BiGuide in an edge-based architecture, using commodity smartphones as mobile clients, and compare it with baseline systems. Extensive experiments demonstrate that OD models trained on the data collected by BiGuide notably outperform those trained on the data collected by two baseline systems, increasing detection accuracy by up to 33.07% and 14.57%, respectively.

ACKNOWLEDGMENTS

We thank Ashley Kwon for her contributions to the project. This work was supported in part by NSF grants CSR-1903136, IIS-2231975, and CNS-1908051, NSF CAREER Award IIS-2046072, Meta Research Award, and Defense Advanced Research Projects Agency Young Faculty Award HR0011-24-1-0001. This paper has been approved for public release; distribution is unlimited. The contents of the paper do not necessarily reflect the position or the policy of the Defense Advanced Research Projects Agency. No official endorsement should be inferred. This is Duke Lemur Center publication #1586. We gratefully acknowledge the contributions of the Duke Lemur Center and the support provided by the Duke Lemur Center's NSF DBI-2012668 Award.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *WIREs: Computational Statistics* 2, 4, 433–459.
[2] Gabriel Aguiar, Bartosz Krawczyk, and Alberto Cano. 2023. A survey on learning from imbalanced data streams: Taxonomy, challenges, empirical study, and reproducible experimental framework. *Machine learning*, 1–79.
[3] Kittipat Apicharttrisorn, Xukan Ran, Jiasi Chen, Srikanth V Krishnamurthy, and Amit K Roy-Chowdhury. 2019. Frugal following: Power thrifty object detection and tracking for mobile augmented reality. In *Proceedings of ACM SenSys*.
[4] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *Proceedings of IEEE ICLR*.
[5] Cenk Baykal, Khoa Trinh, Fotis Iliopoulos, Gaurav Menghani, and Erik Vee. 2022. Robust active distillation. *arXiv preprint arXiv:2210.01213*.
[6] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. 2018. The power of ensembles for active learning in image classification. In *Proceedings of IEEE/CVF CVPR*.
[7] Purnima Bholowalia and Arvind Kumar. 2014. EBK-means: A clustering technique based on elbow method and K-means in WSN. *IJCA* 105, 9.
[8] Mustafa Bilgic and Lise Getoor. 2009. Link-based active learning. In *Proceedings of NeurIPS Workshop on Analyzing Networks and Learning with Graphs*.
[9] Erdem Bıyık, Kenneth Wang, Nima Anari, and Dorsa Sadigh. 2019. Batch active learning using determinantal point processes. *arXiv preprint arXiv:1906.07975*.
[10] Stacy M Branham, Ali Abdolrahmani, William Easley, Morgan Scheuerman, Erick Ronquillo, and Amy Hurst. 2017. "Is someone there? Do they have a gun": How visual information about others can improve personal safety management for blind individuals. In *Proceedings of ACM ASSETS*.
[11] Davide Cacciarelli and Murat Kulahci. 2023. Active learning for data streams: A survey. *Machine Learning*, 1–55.
[12] Pimwadee Chaovalit and Lina Zhou. 2005. Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of IEEE HICSS*.
[13] Jiahong Chen, Tongxin Shu, Teng Li, and Clarence W de Silva. 2019. Deep reinforced learning tree for spatiotemporal monitoring with mobile robotic wireless sensor networks. *IEEE SMC* 50, 11, 4197–4211.
[14] Jian Cheng, Zhiji Zheng, Yinan Guo, Jiayang Pu, and Shengxiang Yang. 2023. Active broad learning with multi-objective evolution for data stream classification. *Complex & Intelligent Systems*, 1–18.
[15] CORe50. 2022. CORe50. https://vlomonaco.github.io/core50/.
[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE/CVF CVPR*.
[17] Alexandre dos Santos Mignon and Ricardo Luis de Azevedo da Rocha. 2017. An adaptive implementation of ϵ-greedy in reinforcement learning. *Elsevier PCS* 109, 1146–1151.
[18] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. 2020. Deep multimodal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE T-ITS* 22, 3, 1341–1360.
[19] Flickr. 2024. Find your inspiration. https://www.flickr.com/.
[20] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *Proceedings of PMLR ICML*.
[21] Javier Garcıa, Daniel Acera, and Fernando Fernández. 2013. Safe reinforcement learning through probabilistic policy reuse. In *Proceedings of RLDM*.
[22] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. 2009. Multi-class active learning for image classification. In *Proceedings of IEEE/CVF CVPR*.
[23] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Proceedings of NeurIPS*.
[24] Michael Laielli, James Smith, Giscard Biamby, Trevor Darrell, and Bjoern Hartmann. 2019. LabelAR: A spatial guidance interface for fast computer vision image collection. In *Proceedings of ACM UIST*.
[25] Guofa Li, Zefeng Ji, and Xingda Qu. 2022. Stepwise domain adaptation (SDA) for object detection in autonomous vehicles using an adaptive CenterNet. *IEEE T-ITS* 23, 10, 17729–17743.
[26] Teng Li, Chaoqun Wang, Max Q.-H. Meng, and Clarence W. de Silva. 2022. Attention-driven active sensing with hybrid neural network for environmental field mapping. *IEEE T-ASE* 19, 3, 2135–2152.
[27] Zenan Li, Xiaoxing Ma, Chang Xu, Jingwei Xu, Chun Cao, and Jian Lü. 2020. Operational calibration: Debugging confidence errors for DNNs in the field. In *Proceedings of ACM ESEC/FSE*.
[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of Springer ECCV*.
[29] Luyang Liu, Hongyu Li, and Marco Gruteser. 2019. Edge assisted real-time object detection for mobile augmented reality. In *Proceedings of ACM MobiCom*.
[30] Weibo Liu, Zidong Wang, Yuan Yuan, Nianyin Zeng, Kate Hone, and Xiaohui Liu. 2021. A novel sigmoid-function-based adaptive weighted particle swarm optimizer. *IEEE T-CYB* 51, 2, 1085–1093.
[31] Shaolong Ma, Yang Huang, Xiangjiu Che, and Rui Gu. 2020. Faster RCNN-based detection of cervical spinal cord injury and disc degeneration. *JACMP*, 235–243.
[32] Mehdi Maboudi, Mohammad Reza Homaei, Soohwan Song, Shirin Malihi, Mohammad Saadatseresht, and Markus Gerke. 2023. A review on viewpoints and path planning for UAV-Based 3D Reconstruction. *IEEE J-STARS* 16, 5026–5048.
[33] Nicole A Maher, Joeky T Senders, Alexander FC Hulsbergen, Nayan Lamba, Michael Parker, Jukka-Pekka Onnela, Annelien L Bredenoord, Timothy R Smith, and Marike LD Broekman. 2019. Passive data collection and use in healthcare: A systematic review of ethical issues. *Elsevier IJMI* 129, 242–247.
[34] Alhassan Mumuni and Fuseini Mumuni. 2022. Data augmentation: A comprehensive survey of modern approaches. *Array*, 100258.
[35] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to measure uncertainty in uncertainty sampling for active learning. *Springer Machine Learning* 111, 1, 89–122.
[36] Mathias A Onabid and Djimeli Tsamene Charly. 2017. Enhancing gray scale images for face detection under unstable lighting condition. *SAI IJACSA* 8, 10.
[37] Giorgia Pasini. 2017. Principal component analysis for stock portfolio management. *IJPAM* 115, 1, 153–167.
[38] Eva Patel and Dharmender Singh Kushwaha. 2020. Clustering cloud workloads: K-means vs Gaussian mixture model. *Elsevier PCS* 171, 158–167.
[39] Matthias Perkonigg, Johannes Hofmanninger, Christian Herold, Helmut Prosch, and Georg Langs. 2022. Continual active learning using pseudo-domains for limited labelling resources and changing acquisition characteristics. *MELBA* 7, 1–27.
[40] PyTorch. 2023. Torchvision object detection finetuning tutorial. https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html.
[41] Jiongming Qin, Cong Wang, Qinhong Zou, Yubin Sun, and Bin Chen. 2021. Active learning with extreme learning machine for online imbalanced multiclass classification. *KBS* 231, 107385.
[42] Qualtrics XM. 2023. Qualtrics XM: The leading experience management software. https://www.qualtrics.com/.

[43] Kjetil Raaen, Ragnhild Eg, and Carsten Griwodz. 2014. Can gamers detect cloud delay. In *Proceedings of IEEE NSSG*.

[44] Yatharth Ranjan, Zulqarnain Rashid, Callum Stewart, Pauline Conde, Mark Begale, Denny Verbeeck, Sebastian Boettcher, Richard Dobson, Amos Folarin, and RADAR-CNS Consortium. 2019. RADAR-base: Open source mobile health platform for collecting, monitoring, and analyzing data using sensors, wearables, and mobile devices. *JMIR mHealth and uHealth* 7, 8, e11734.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of NeurIPS*.

[46] Walter Ritter, Guido Kempter, and Tobias Werner. 2015. User-acceptance of latency in touch interactions. In *Proceedings of UAHCI*.

[47] Marieke Rohde, Meike Scheller, and Marc O Ernst. 2014. Effects can precede their cause in the sense of agency. *Neuropsychologia* 65, 191–196.

[48] Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of IEEE ICLR*.

[49] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. 2020. Deep active learning: Unified and principled method for query and training. In *Proceedings of AISTATS*.

[50] Ke Tong, Chad Dubé, and Robert Sekuler. 2019. What makes a prototype a prototype? Averaging visual features in a sequence. *Springer APP*, 1962–1978.

[51] Thomas Waltemate, Irene Senna, Felix Hülsmann, Marieke Rohde, Stefan Kopp, Marc Ernst, and Mario Botsch. 2016. The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. In *Proceedings of ACM VRST*.

[52] Tao Wang, Xiaopeng Zhang, Li Yuan, and Jiashi Feng. 2019. Few-shot adaptive Faster R-CNN. In *Proceedings of IEEE/CVF CVPR*.

[53] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. 2023. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of IEEE/CVF CVPR*.

[54] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. 2022. Freesolo: Learning to segment objects without annotations. In *Proceedings of IEEE/CVF CVPR*.

[55] Yongyong Wei and Rong Zheng. 2021. Multi-robot path planning for mobile sensing through deep reinforcement learning. In *Proceedings of IEEE INFOCOM*.

[56] Wikimedia Commons. 2024. Wikimedia Commons. https://commons.wikimedia.org/wiki/Main_Page.

[57] Shuyin Xia, Daowan Peng, Deyu Meng, Changqing Zhang, Guoyin Wang, Elisabeth Giem, Wei Wei, and Zizhong Chen. 2020. A fast adaptive K-means with no bounds. *IEEE T-PAMI*, 1–1.

[58] Ran Xu, Chen-lin Zhang, Pengcheng Wang, Jayoung Lee, Subrata Mitra, Somali Chaterji, Yin Li, and Saurabh Bagchi. 2020. ApproxDet: Content and contention-aware approximate object detection for mobiles. In *Proceedings of ACM SenSys*.

[59] Yi Xu, Asaf Noy, Ming Lin, Qi Qian, Hao Li, and Rong Jin. 2020. Wemix: How to better utilize data augmentation. *arXiv preprint arXiv:2010.01267*.

[60] Xuyang Yan, Mrinmoy Sarkar, Benjamin Lartey, Biniam Gebru, Abdollah Homaifar, Ali Karimoddini, and Edward Tunstel. 2023. An online learning framework for sensor fault diagnosis analysis in autonomous cars. *T-ITS* 22, 3, 1341–1360.

[61] Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. 2017. Deep similarity-based batch mode active learning with exploration-exploitation. In *Proceedings of IEEE ICDM*.

[62] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. 2017. Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge. In *Proceedings of IEEE ICRA*.

[63] Xueying Zhan, Qing Li, and Antoni B Chan. 2021. Multiple-criteria based active learning with fixed-size determinantal point processes. *arXiv preprint arXiv:2107.01622*.

[64] Xianghao Zhan, Yuzhe Liu, Nicholas J Cecchi, Olivier Gevaert, Michael M Zeineh, Gerald A Grant, and David B Camarillo. 2022. Finding the spatial co-variation of brain deformation with principal component analysis. *IEEE T-BME*, 3205–3215.

[65] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. 2022. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*.

[66] Bingqing Zhang, Sen Wang, Yifan Liu, Brano Kusy, Xue Li, and Jiajun Liu. 2023. Object detection difficulty: Suppressing over-aggregation for faster and better video object detection. In *Proceedings of ACM Multimedia*.

[67] Han Zhang, Yucong Yao, Ke Xie, Chi-Wing Fu, Hao Zhang, and Hui Huang. 2021. Continuous aerial path planning for 3D urban scene reconstruction. *ACM TOG* 40, 6, 225–1.

[68] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. 2021. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In *Proceedings of IEEE ICCV*.

[69] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. 2021. Adversarial attacks against LiDAR semantic segmentation in autonomous driving. In *Proceedings of ACM SenSys*.