LabelAR: A Spatial Guidance Interface for Fast Computer Vision Image Collection

Michael Laielli*, James Smith*, Giscard Biamby*, Trevor Darrell, Bjoern Hartmann
UC Berkeley EECS, Berkeley, CA, USA
{laielli, james.smith, gbiamby, trevor, bjoern}@berkeley.edu

ABSTRACT
Computer vision is applied in an ever expanding range of applications, many of which require custom training data to perform well. We present a novel interface for rapid collection and labeling of training images to improve computer vision-based object detectors. LabelAR leverages the spatial tracking capabilities of an AR-enabled camera, allowing users to place persistent bounding volumes that stay centered on real-world objects. The interface then guides the user to move the camera to cover a wide variety of viewpoints. We eliminate the need for post-hoc manual labeling of images by automatically projecting 2D bounding boxes around objects in the images as they are captured from AR-marked viewpoints. In a user study with 12 participants, LabelAR significantly outperforms existing approaches in terms of the trade-off between model performance and collection time.

Author Keywords
spatial interfaces; augmented reality; computer vision; image collection

CCS Concepts
•**Human-centered computing** → **Mixed / augmented reality;** *User interface design;* User studies;

INTRODUCTION
Computer vision is being used in an increasing number of user-facing systems. Deep neural networks used in modern computer vision can require copious amounts of training data. Modern applications often rely on large datasets (10-200GB) to train these models. Often these datasets are collected by scraping the internet for existing images and then manually labeling them, for example through crowdsourcing [9].

However, existing datasets produced in this fashion do not cover the "long tail" of use cases — where users have a need to detect specific classes of items that are not already labeled in existing datasets. An example would be that we may want to ask a robot to "bring me my jacket." Although there is plenty of training data of jackets, there are no efficient ways to collect training data for a specific instance of a category ("my jacket"). Another example is needing a vision system to distinguish between different types of industrial hardware. Datasets for such fine grained categories can be difficult to source due to the effort needed to collect them.

Existing approaches to overcome this bottleneck include parallelizing the labeling task with post-hoc annotation tools [38] or simple camera interfaces to guide image collection such as putting a bounding box in a viewfinder to avoid post-collection labeling [15]. The first approach can produce high quality data but is a very time intensive process that scales linearly with the number of objects that have to be labeled. The second approach is significantly faster, but produces lower quality data, as the bounding box accuracy is compromised.

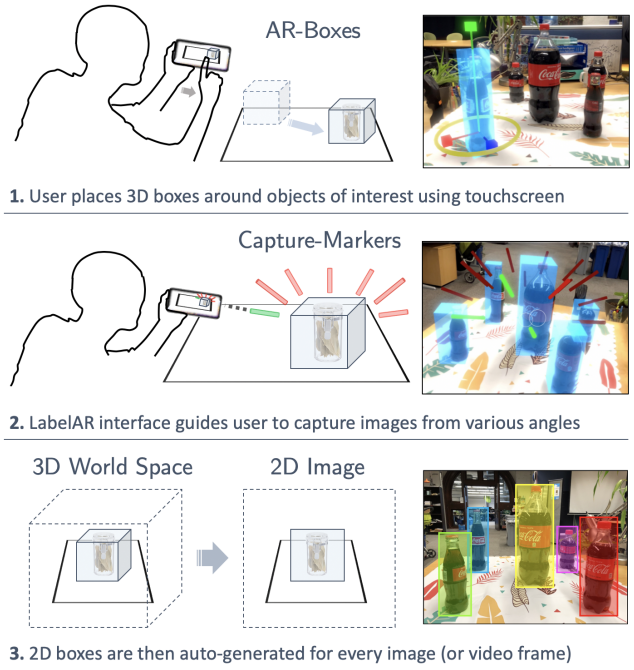

Figure 1. LabelAR uses augmented reality to speed up and structure the in-situ collection of images and labels for training object detectors.

* Equal contribution.

UIST'19, October 20–23, 2019. New Orleans, LA, USA

ACM ISBN xxx-x-xxxx-xxxx-x/xx/xx.
DOI: http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx

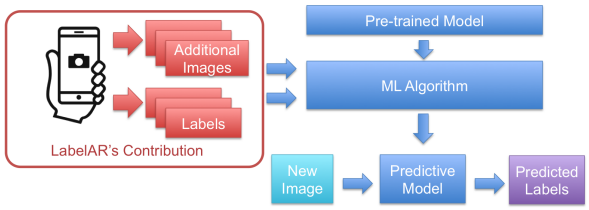

Figure 2. LabelAR produces additional training data to adapt computer vision models to detect additional objects in a user's environment.

We propose LabelAR, an augmented reality interface that allows users to rapidly collect and label high quality training image datasets for computer vision (see Figure 1). LabelAR is applicable to any setting where a user needs to adapt a computer vision object detection model (see Figure 2). Transfer learning is a technique that can leverage a model pre-trained on large datasets to recognize new objects. Training data needed for transfer learning needs to be situated in context and diverse in viewpoint variety.

Two illustrative use cases where such adaptation is necessary are *augmented reality assembly* and *home robotics*. In *augmented reality assembly*, a worker employs a head-mounted AR device to project visual, step-by-step instructions that demonstrate how to assemble a collection of object parts. Object detection can be used for identifying and tracking particular parts throughout the assembly. A recent study concluded that better tracking capabilities are still needed for AR assembly to be sufficiently robust for industrial applications [11]. In *home robotics*, a robot owner would adapt a computer vision model for use on a robotic platform to recognize individual items in a household that may be significantly different from items in existing training sets.

Our interface design is based on the observation that computer vision model performance depends on the quality of the training data. Two important aspects of high quality training data are accurate bounding boxes for labels and a diversity of images (orientations, scales, etc.) of the objects to be recognized [5]. Our interface embodies the following two insights:

First, AR spatial tracking enables users to quickly and accurately place spatially stable 3D bounding volumes around objects in their environment, which can be automatically transferred to 2D bounding boxes at any camera angle (Figure 1).

Second, interactive guides in the AR interface can prompt the user to collect training data instances of the bounded objects from many viewpoints. We show that these appearance variations combined with accurate bounding box labels improve detection accuracy.

Through empirical experimentation, we show that our interface enables a better trade-off between time costs and model performance than existing baseline methods. We conduct a user study with 12 participants that compares collection times and model performance (when trained on collected images) between LabelAR and two alternative interfaces [38][15]. Compared to post-hoc labeling (with free-form camera), collection times improved by over 2× on average with LabelAR ($p<0.001$), while model performance was similar. Compared to an existing rapid collection application, model performance increased by a factor of 5× on average ($p<0.01$), while collection times increased only by 27%. Additional empirical analysis shows that the equally spaced viewing angle intervals afforded by LabelAR are effective for low-sample learning and fast computer vision model training.

The main contributions of this work are the design, implementation and evaluation of an augmented reality interface for fast computer vision image collection. We demonstrate both AR-capable smartphone or head-mounted device implementations and show gains in collection time and model performance over existing approaches.

RELATED WORK

Prior work falls in the areas of image collection methods, post-hoc labeling interfaces, and other spatial user interfaces for interacting with 3D content. We review each area in turn.

Image Collection

Several projects seek to shorten or eliminate post-hoc labeling time through novel capture-time interfaces and techniques. Raptor [15] modifies the camera interface by overlaying a pre-sized bounding box. The user points it to a new object, ensuring it is displayed within the pre-defined box area. Since the object is fit to the box with known image coordinates, there is no subsequent labeling task required. However, this results in images that are all collected at the same scale. The Doubleshot technique [37] asks users to take two images, one with the object, one without the object (my manually removing it) to automatically calculate labels.

Recent work at the intersection of cognitive development and computer vision shows deep neural network categorization performance can be improved by increasing the variety of viewing angles at which training images are collected. [5]. We seek to exploit AR to guide the user to capture such a variety of images.

Sermanet et al. [30] use a two-person collection strategy for capturing multiple views of an object simultaneously. This allows training to be "self-supervised". Our work differs in that it uses spatial tracking and user-placed 3D bounding volumes to create labeled images.

Another set of collection approaches use head-mounted cameras to collect video along with separate audio recording devices to allow the user to speak the labels verbally, either simultaneously [33] or immediately after video collection [7]. Both of these approaches rely on speech recognition APIs to extract functional labels from the user recordings. While these prove effective for categorization labeling, they do not aid in the placement of bounding box labels.

Other approaches employ robots to perform the image collection [34, 21, 25]. The Amazon Robot Picking Challenge shows joint collection and labeling applied to a challenging real-world task [39]. They use a robot arm to capture multiple angles of a single object and, with knowledge of the background, automatically obtain segmentation labels by foreground masking. The idea behind LabelAR is similar, but

our application guides people to collect the images. Furthermore, LabelAR is applicable to more than one object at a time and does not rely on prior knowledge of the background appearance.

There are a few interfaces that guide user viewing angle. Vuforia PTC provides a single-object capture interface for 3D model construction [18]. Ours is a multi-object capture interface, which is crucial for the purpose of deep learning based multi-object detection. Google Photo Sphere is a panoramic image capture interface that guides users to orient their phones according to an inside-out, spherical collection structure [23]. Ours differs by guiding the user to look inward toward a bounded group of objects. We take inspiration for the alignment interface design from Photo Sphere.

Interfaces for Post-Hoc Labeling

Image and video annotation for computer vision model training and adaptation is typically done via web interfaces where a human annotator sits at a terminal and uses a mouse to draw bounding boxes or segmentation boundaries on various objects of interest [19].

Web-based 2D bounding box labeling interfaces such as LabelMe [29] have been integral in constructing some of the most influential computer vision data-sets to date [12, 9], allowing labeling tasks to be distributed in the form of global crowdsourcing campaigns.

There are a few works that use 3D labeling tools to label a 3D scene, then leverage the 3D labels to generate large amounts of 2D labels [35, 6]. Our work makes use of 3D to 2D label transfer, but for real-time collection of interactive objects rather than post-hoc passive labeling of 3D scenes.

Several works leverage interactivity between the learner and predictive model to reduce human labeling time and effort by having the model predict labels that can then be approved or improved by the user [3]. *Crayons* [13] uses a simple interactive painting metaphor to reduce classifier creation time. A more recent approach incorporates interactivity into web-based crowdsourcing tools showing that interactive modes can reduce the number of annotator mouse-clicks by as much as 50 percent [1]. *CueFlik* [2] presents a design and evaluation of new approaches to guiding users in selected training examples interactively based on model predictions. Our work does not incorporate interactivity in this sense, rather it focuses on the interaction between the user and real-world objects of interest. *Eye-patch* [24] is a tool for designing camera-based interactions. The authors identify a need to accelerate the example-collecting process as a result of their deployment, which is aligned with LabelAR's goals.

Spatial User Interfaces for Interacting with 3D Objects

Technologies and interaction techniques for spatially tracked screens and near-eye displays have been a focus of HCI research since pioneering efforts by Sutherland [31], Fitzmaurice [14] and others. A number of spatial interaction techniques can now be found in the literature — e.g. in surveys by Hinckley [16] and Argelaguet Sanz [4]. Early interfaces such as Peephole Displays [36] and the Boom Chameleon [32] used translational and rotational tracking of a hand-held display to navigate, view and annotate with large virtual maps and 3D models, respectively. Most relevant to LabelAR are interfaces for *Situated Modeling* where real-world context is used to create and place 3D geometry such as our bounding volumes [20, 17]. Our contribution differs in that our created 3D geometry is a means towards the end of collecting image sets and we study the benefits of such an approach for computer vision.

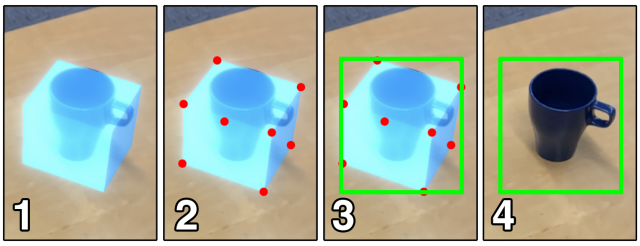

Figure 3. Process for extracting two dimensional bounding boxes from virtual bounding volumes. 1 - a virtual bounding volume is placed over a real object, 2 - the corners (red) of the volume are projected into camera space, 3 - a min/max is taken over those points to find a two dimensional bounding box (green), 4 - annotations are saved for the object.

LABELAR INTERACTION DESIGN

At a high level, LabelAR seeks to lower the time that a user needs to spend collecting and annotating images of objects for training neural networks. We identified two key areas for improvement in existing workflows.

Hand annotating bounding boxes: hand annotated bounding boxes are often pixel perfect and very high quality, but take a long time to author. Today, this process is sometimes parallelized through crowdsourcing, which reduces time but not the total amount of labor required. We hypothesize that AR technology can automate the process of producing bounding boxes to a high degree, given an initial 3D bounding volume of an object.

Capturing a large variety of training examples: large training data sets are needed to train computer vision models. We hypothesize that offering a guided experience with feedback to the user about how much of a variety they have collected can help them produce smaller training sets that will result in better model performance.

Placing 3D Bounding Volumes

In existing workflows, when annotating a series of images that contain the same objects, users have to annotate a given object multiple times. LabelAR speeds up this process by asking the user to place a bounding volume around an object once, and then tracks it in all subsequent images.

To this end, we utilize AR technology. One of the important technologies in AR devices is the ability for them to self-localize, e.g. using SLAM [8]. This allows virtual objects to be placed in the real world environment, and for their positions and orientations to remain coherent.

LabelAR allows users to place virtual bounding boxes (holocubes) over objects in the environment so their positions

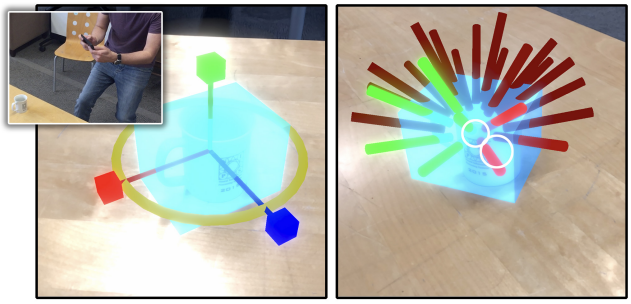

Figure 4. Left: Users can move, re-size, and rotate bounding volumes by interacting with the TRS widget. Right: Markers indicate where good potential images should be taken from next, and change color to indicate the user's proximity to the correct location (red) or if they have already captured an image at that location (green). Images are automatically captured when the user is in position and the two targeting circles align.

can be tracked. When an image is taken in our interface, the two-dimensional bounding box can be computed for every object in the scene by projecting the holocube into the video frame and finding the bounding box of its vertices (Figure 3). It is thus important that the holocubes fit the size and shape of the objects of interest as closely as possible. To this end, we provide users translation, rotation, and scaling (TRS) widgets to manipulate the 3D position and size, as well as rotation around the axis normal to ground plane of the holocubes. The interaction design for the TRS manipulators mimics conventional widgets in 3D graphics software (Figure 4).

Users can also place cubes over multiple objects. In traditional approaches, the labeling effort grows as the product of $images \times objects$, which becomes prohibitive for long sequences containing many objects. In LabelAR, each additional object to be captured only incurs the one-time effort of placing and adjusting another bounding volume.

Encouraging Diverse Image Perspectives

LabelAR helps users collect a wide variety of images of given objects. In particular, it directs users to capture objects from many azimuth and altitude angles. To accomplish this, we provide an interface to assist users in taking pictures of their objects. LabelAR will automatically take a picture if the user has a significantly different viewpoint than all previous pictures taken. This encourages the user to move the camera around the objects they wish to capture to ensure they get a variety of angles.

To facilitate this movement, our interface shows the user where they have already taken pictures and suggests new positions at which to take pictures while also requiring them to keep the holocubes in frame.

We visualize this constraint to the user as a series of rods that we refer to as markers (Figure 4). Rods were chosen as they are the simplest shape that indicates a directional 3D vector. Markers change their color as users approach the correct position, providing them with real-time feedback about their progress. There is also a targeting cursor to direct the camera's orientation (Figure 4). A target for the cursor appears

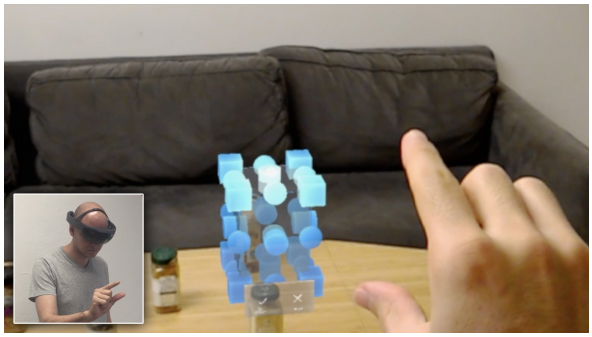

Figure 5. The platform-specific translate, rotate, scale widget in the HoloLens version of the interface. (Sub-picture shows interface in use)

at the location in space that the user should point the camera to. Images are taken for the user automatically as soon as they are in position and the targeting cursors are aligned. The rods turn from red to green to notify the user when and where they have captured images.

IMPLEMENTATION

We implemented LabelAR on two different AR platforms: hand-held, video-see-through on iOS devices, and head-mounted AR using Microsoft's HoloLens. Both versions were developed in the Unity game engine. A custom application was written to perform the function of LabelAR, and platform APIs were used to tie into hardware-specific features. In development, we tested LabelAR on the HoloLens, iPad Pro, and iPhone. With some sessions exceeding 1 hour, there was noticeable fatigue with the HoloLens due to hand gestures and head pointing, and with the iPad which was heavy to hold with one hand while the other manipulated the holocubes. We did not experience fatigue with the iPhone, and although there is substantial waving, we found no strains of the shoulder as with HoloLens hand gesturing. Therefore we selected the iPhone interface as the only interface tested for the study.

The iOS version of LabelAR uses Apple's ARKit library, in particular the camera localization and plane detection functionality. As surfaces in the environment are detected, they are converted into planes that can be utilized in a Unity application. When the user places holocubes, they are automatically snapped to surfaces by performing raycasts onto these planes. To create new cubes, the user can press and hold a "new object" button that hovers a cube in front of them, which they can drop onto a table. Interaction with the holocubes and interface happen with the device's touch screen. When running ARKit apps on iOS, the camera is put into a special video mode that has a different field of view than the standard camera mode. Because of this, images are saved directly from the Unity rendering pipeline to make sure that the projected bounding boxes are guaranteed to line up. We downsample these images before saving to improve serialization times, and because many cv model training pipelines downsample training data. Images are saved at 960×540 resolution, and annotations are saved in the COCO JSON format [22]. Images and annotations are automatically saved to the persistent storage of the application, which can be downloaded through XCode. This was chosen

instead of saving images to the iOS Photos app so that images and annotations could be co-located.

The HoloLens version (Figure 5) uses Microsoft's Mixed Reality Toolkit (MRTK), which contains a set of assets for building native HoloLens applications within Unity. Interactions on the HoloLens are performed by using the standard gaze plus thumb-and-index-finger pinch gesture. MRTK contains a set of widgets for manipulating the size and orientation of virtual objects, so these were used in lieu of our own TRS widget. Objects are moved by using the built in hand tracking detection native to the hardware. Interacting with screen space interface elements has some issues on head mounted AR devices because the interactions are gaze driven, so we opted to use the phrase detection and dictation engines available on the hardware to allow the user to place new holocubes and initiate or stop the capture process. Images are saved at 1280×720 pixels. Captured images and bounding boxes are sent over a network connection to a server for collection, and bounding box data is converted into COCO JSON format.

EVALUATION

We conducted a user study along with supporting ablation experiments to answer the following questions:

1. **Is AR based data collection faster?** How much time does it take to collect and annotate images compared to existing baseline methods?
2. **Does our AR interface result in better object detection models?** Does training on data collected via our interface result in a better model than training on data captured with other baseline collection-and-labeling methods?
3. **How accurate are the labels produced by AR-based image collection?**
4. **How sensitive is detection performance to design choices such as the number of angles presented in the guidance interface?**

Our goal in defining an experimental setup was to portray a challenging object detection task in an environment that is both realistic and representative of plausible of use-cases. We constrain the problem to the computer vision task of multiple object detection: That is the joint task of categorizing and localizing (via bounding boxes) any instances of a predefined set of objects. This is the type of problem that would need to be solved for the motivating *in home robotics* and *augmented reality assembly* scenarios.

Variables

The primary **independent variable** in our study is the interface used to capture and label images. We compare LabelAR to two baseline methods: free-form image capture with post-hoc annotation using the Scalabel tool [38]; and 2D guided capture using using an overlayed 2D bounding box collection tool, such as the one in the Raptor project [15]. Thus we have one independent variable with three levels: 1) LabelAR; 2) Post-hoc annotation; and 3) overlayed bounding box interface (referred to henceforth as "post-hoc" and "overlayed" interfaces).

The two primary **dependent variables** in our comparative study are *collection time* for an image set and object detection *model performance* when trained on collected images. We define collection time as the total time required for a participants to capture and completely label a set of images for 5 objects, each from multiple angles.

Collected image sets are used to train Faster R-CNN models [26] which are then evaluated on hold-out test sets. We define detection performance as the average precision (AP) at a given intersection-over-union (IOU) threshold. IOU is the area of intersection of two boxes (predicted vs. target) divided by the area of their union. AP is a standard metric for object detection in popular computer vision benchmarks [22]. We chose to investigate IOU thresholds of 0.25 and 0.5.

We also investigate bounding box accuracy on *collected* images by calculating IOU between sets of randomly chosen participant-collected images and gold-standard labels created by researchers. Finally, we collect qualitative feedback through a post-study survey.

Participants

12 participants were recruited through email invitations sent to the departmental list-serves of the EE and CS departments at our university. The mean age was 26.4 years. 9 of the participants were male and 3 were female. Half of the participants had at least some prior experience with machine learning, computer vision, or both. All participants had taken classes in computer science. Although our study audience all have a technical background relative to the general population, we believe that this makes them well suited to perform better in using the non-LabelAR interfaces as they likely have more prior understanding how vision algorithms work. We also recognize that our study participants have high levels of technical literacy that will likely make all tested interfaces perform better than the general population average.

Apparatus

Each study took place in the same office space with one participant and one researcher administering the experiment. Each participant performed three separate collection-plus-labeling tasks using a dedicated app for each task on an iPhone 8plus. Each task asked the user to capture a series of images of multiple objects placed on a round table in the center of the room. The object sets were switched out between tasks so that no user had the same set twice. The table was covered with a patterned table cloth to ensure the ARKit low-level functions had a sufficient amount of visual features for stable plane detection and tracking. All three apps were custom built with Unity and ARKit 2.0, and were deployed to the same iPhone 8plus running iOS 12.2 to ensure that all images were recorded in the same encoding, resolution, and aspect ratio (*sRGB*, 960 x 540) as the evaluation set. The Scalabel interface was run on a 2017 15-inch MacBook Pro running macOS Mojave version 10.14.2. The participants were given the choice of using a mouse or the laptop track pad for drawing bounding box labels.

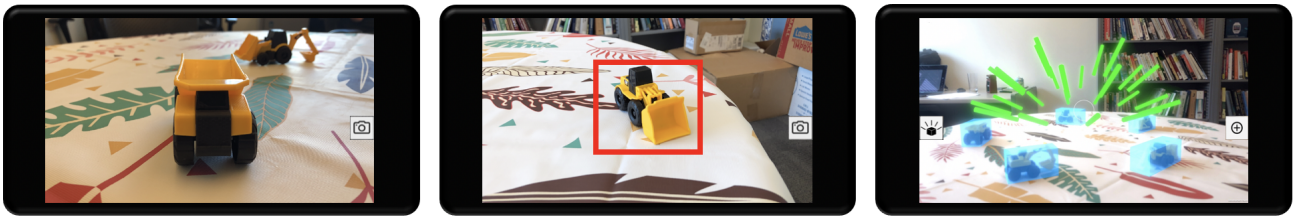

Figure 6. Depictions of the free-form camera (post-hoc), overlayed, and LabelAR interfaces (left to right) with actual screen shots.

Procedure

Upon receiving user consent, the researcher gave a 2-min. project intro starting with the following description:

> Let's say you bought a robot that cleans up your room. You want it to detect your personal items so it knows where to put them. LabelAR would help you teach the robot to detect those things. So, you're going to take a bunch of pictures of (these) items laid out on the table with a few different apps.

The study purpose and procedural overview were then explained.

The first task was to use the free-form camera app to collect images of an initial set of objects (Figure 6). A 1-min tutorial was given on a practice item to familiarize the participant. The initial set of five objects were then introduced. A few suggestions, consistent with common computer vision best practices were made:

1. *Take images from varied viewpoints.*
2. *Balance the number of times each object appears among the collected images and make sure each object appears at least a few times.*

Participants were given a time limit of seven minutes to take as little or as many pictures as they want. They were told that the robot would need to recognize objects from various locations around a room and that they can move the objects around the table if they want, but not to bother flipping the object sideways or upside-down.

The second task proceeded much like the first, with the overlayed interface (Figure 6). The participant was advised to take images of only one object at a time and to ensure the object fit in the box on the screen without exceeding the boundaries and without appearing too small. The researcher explained the other objects on the table should not appear within the box or elsewhere in the image. The same guidance on viewpoints, time limit, and number of images was given as in the first task. A new set of 5 objects was placed on the table.

The third task used our interface. The researcher opened and initialized the app before handing it over by scanning the table for a few seconds to let ARKit find low-level visual features. The participant was advised to keep the phone generally pointed towards the table so that the app doesn't lose track of the table position. The participant was guided to place and fit an AR-box over a practice object, first fitting the sides of the object by positioning for a top-down view, then adjusting box-height from a side view. Then, the participant was told how to activate the capture marker system and how to capture an individual marker. Similar to the other tasks, a new set of five objects was placed in a rough circle on the table. The participant was told they should not move the objects once the capture markers were activated. Participants did receive advice on viewpoints or number of images, rather just to ensure each marker turns green.

The post-hoc labeling phase was to use the Scalabel tool to label the images collected with the free-form camera app (the first task). A 5-minute tutorial was given on how to categorize and draw 2D boxes around objects in the images in a time-efficient manner. The participant was advised to label overlapping objects to the extents of their respective visual features. Similarly, if an object was truncated by the image boundary, the participant was advised to label it only if 30% or more of the object was visible.

In our evaluation, tasks were always conducted in order of increasing guidance. While this choice might create an ordering effect, we selected this design to minimize transference and bias from one task to the next. In particular, we wanted to elicit user's un-aided image collection behavior first. We hypothesize that experiencing an interface that guides participants to capture a sufficient amount of out-of-plane rotations might create a strong bias for all subsequent interfaces, precluding counterbalancing the order.

Analysis

In our experiment, we focus on the computer vision task of object detection where the goal is to categorize and locate objects by placing bounding boxes around them. For each set of training images collected by the user study participants, we train a Faster R-CNN [26] detection model until convergence. These detection models were then evaluated on one of three hold-out test sets of 120 images that feature the objects from the respective user collection, scattered about desks and floors in new environments unseen during the training phase. Roughly half of the test images feature cluttered scenes, with some containing occlusions. Each object instance in the test set was meticulously labeled with a 2D bounding box. Some sample test images are shown in Figure 7.

We ran our experiments with three sets of objects: Cola bottles, toys, and industrial hardware. Each set consists of five distinct instances, hand picked for categorical granularity finer than a typical category in the ImageNet-1000 set [27]. Thus, a detector that is only trained on the ImageNet or COCO dataset would fail without additional training images. The objects were also chosen to ensure a variety of sizes and shapes: the

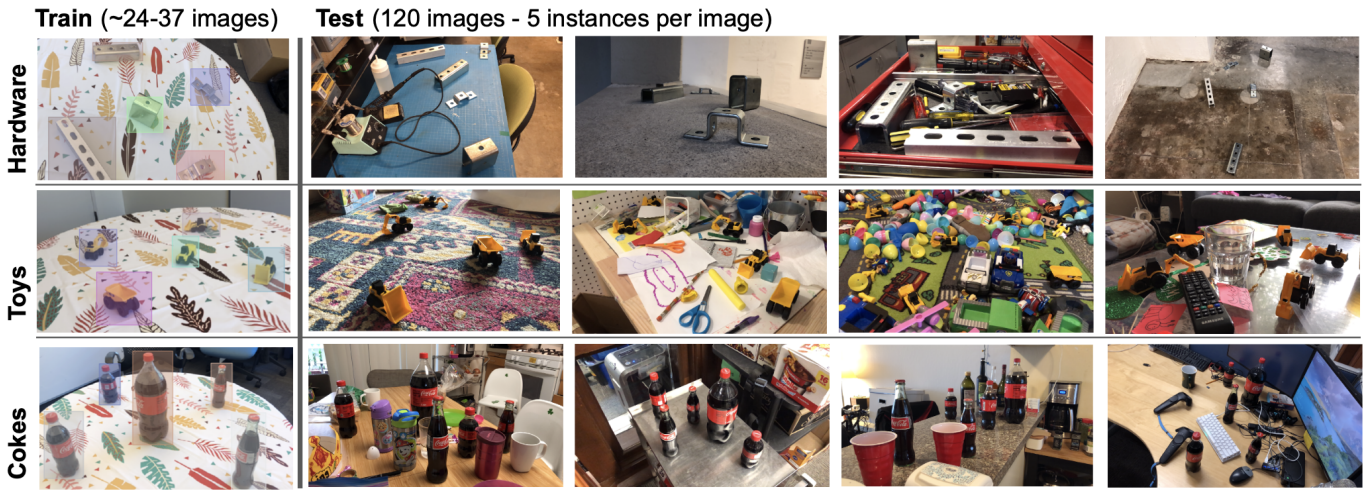

Figure 7. Samples from train-test combinations (row-wise) demonstrate the challenging generalization task posed by our experiments. All training images were collected in the same room while test images were collected in new rooms, configured for variations in lighting, scale, clutter, and occlusions.

		Object detection performance	
Interface	*Collect time (min)*	*mAP .25IOU*	*mAP .5IOU*
Post-hoc	15.97	0.40	0.32
Overlayed	4.82	0.10	0.02
LabelAR	6.11	0.58	0.22

Table 1. Results of detection models trained on user-collected images and tested on images of the same objects in new scenes.

bottles are tall, the hardware is flat, and the toys are small. Since any level of desired detection performance is a function of IOU accuracy and object size, the small objects also add rigor to our evaluation by testing robustness to tracking inaccuracies. In the case of bounding a toy object ($\sim$3x2x2cm), a shift of $\sim$1cm could result in an IOU loss of up to -0.66. This is unacceptable if an IOU threshold of 0.5 is needed.

We constrained our experiment environment to static objects placed on a table. This *multiple-objects-on-a-table* setup is common across cognitive development literature, robot learning (e.g. visual pick and place tasks), and fits with a common AR use case of multi-object assembly.

RESULTS

We first present and discuss quantitative results of our user study, then review qualitative findings.

Quantitative results

Compared to post-hoc labeling, collection times improved by over 2x with LabelAR, while model performance did not show a statistically significant difference. Compared to the existing rapid collection application, model performance increased by a factor of 5x on average.

Figure 8 shows collection times and average precision (AP) at 0.25 IOU and 0.5 IOU for all users and the three tested interfaces. These graphs allow us to investigate the trade-off between collection time and object detection performance.

Table 1 summarizes the mean values for collection time and average precision across all users. A 0.5 IOU threshold was chosen for our primary evaluation since it is well accepted by the computer vision community [28]. A 0.25 IOU threshold could be used for applications that are less dependent on localization accuracy e.g. a key-finder app. None of the interfaces produce usable detectors at 0.75 IOU (average AP for each interface is below 0.1).

Collection Time

Repeated measure ANOVAs showed that for log-transformed completion times, ANOVA showed a significant effect of interface ($F(2,22)=154.8$, $p<0.001$) and the post-hoc pairwise t-tests with Bonferroni correction were significant between all three pairs of groups. Collecting labeled images with LabelAR ($\mu = 6.11$ min) is significantly faster ($p<0.001$; by 9.9 min or $2.6\times$ faster) than using post-hoc annotation ($\mu = 15.97$ min). LabelAR is significantly slower ($p<0.01$; by 1.3 min or $1.27\times$) than the overlayed interface ($\mu = 4.82$ min)

Average Precision

Repeated measures ANOVA showed a significant effect of interface on AP at IOU thresholds of 0.25 ($F(1.37, 15.07)=39.00$, $p<0.001$) and 0.5 ($F(2, 22)=11.03$, $p<0.001$). The post-hoc pairwise t-test showed that at an IOU threshold of 0.25 there was not a significant difference in detection performance between LabelAR ($\mu = 0.58$) and Post-hoc labeling ($\mu = 0.40$), but the detection performance of LabelAR compared to the overlayed interface ($\mu = 0.10$) was significantly better ($p<0.001$). The post-hoc tests for the AP 0.50 threshold did not find a significant performance difference between LabelAR ($\mu = 0.22$) and the post-hoc annotation interface ($\mu = 0.32$), while the LabelAR detection performance was significantly better ($p<0.01$) than the overlayed interface ($\mu = 0.02$).

To summarize, LabelAR was more than twice as fast in collecting images and labels than post-hoc annotation, while its associated object detection performance was comparable to that of post-hoc annotation. LabelAR is somewhat slower to

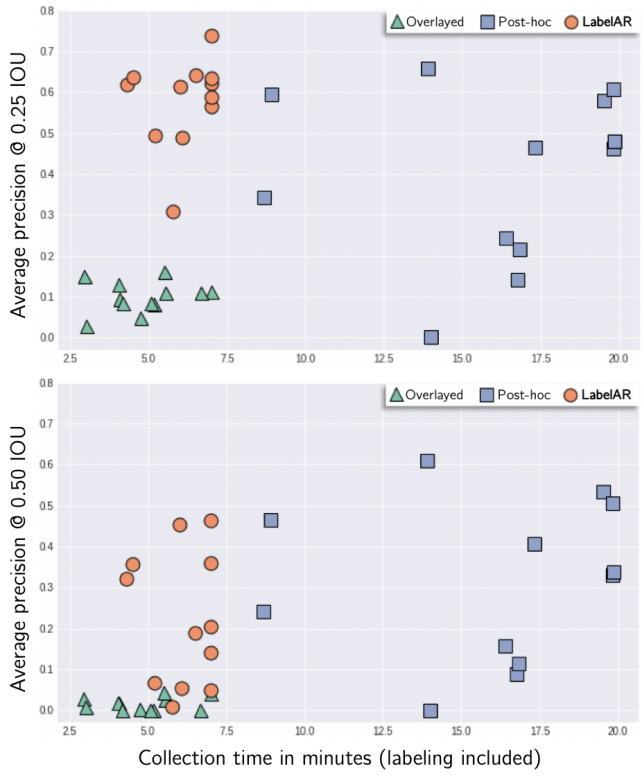

Figure 8. Results for object detection performance versus collection time is shown at two IOU thresholds.

use than 2D overlayed bounding boxes (by 1.3 minutes on average). That difference is much smaller than the difference to post-hoc annotation. LabelAR significantly outperforms overlayed bounding boxes in object detection by at least a factor of 5× at both investigated IOU levels.

Bounding Box Accuracy of Collected Images

We compared the accuracy of bounding boxes on objects in collected images produced by participants against a meticulously labeled "gold-standard" set for randomly chosen participant-collected images in terms of intersection-over-union (IOU). 100 images for each condition were sampled at random, and a single annotation per image was hand annotated and then analyzed against the annotation produced during the study. The prior section used IOU of *predicted* versus gold-standard boxes on holdout *test* images, while this analysis focuses on *collected bounding boxes* versus gold-standard boxes on participant-collected *training images*.

A one-way repeated-measures ANOVA revealed a statistically significant effect of interface on IOU ($F(2, 22)=59.46$, $p<0.001$). A post-hoc pairwise t-test with Bonferroni correction revealed statistically significant ($p<0.05$) differences between LabelAR ($\mu = 0.50$) and the overlay interface ($\mu = 0.45$). The mean IOU for the post-hoc interface ($\mu = 0.90$) was significantly higher ($p<0.001$) than for LabelAR.

Number of Images Taken

The results we obtained are dependent on the number and variety of images participants take in different conditions. While

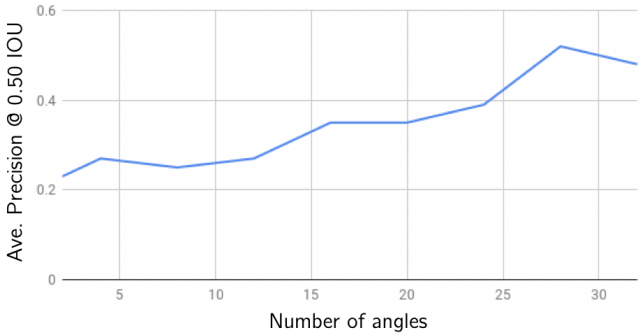

Figure 9. Object detection performance versus the number of images used in training (collected at maximally spaced viewing angles).

LabelAR heavily guides users, the other conditions offer little-to-no guidance. We investigate how many images participants took organically across three conditions here, then investigate how sensitive LabelAR performance is to the number of images and angles captured in the next section.

On average, users took 29.8 ($\sigma = 14.8$) images using the free-form camera app and annotated anywhere from 1-5 objects per image in the post-hoc annotation tool. Users took an average of 37.0 ($\sigma = 24.8$) images using the overlayed interface, with at most 1 annotation per image. Users took exactly 24 images in LabelAR, as that is the number of images suggested by our guided interface, each image having 5 annotations.

Angle Ablation for LabelAR

How many different angles should LabelAR ask a user to collect? To drive design decisions of the LabelAR interface, we performed an angle ablation on the coke bottles object set, starting with 32 angles and decreasing to 2 by increments of 4 (and by 2 on the last one). Figure 9 shows a linearly increasing relationship between the number of angles and detection performance, up to 32 equally space horizontal angles. That is, there is no "knee" in the curve at which point adding angles has a diminishing return on investment in terms of object detection performance. For this evaluation set, the more angles we capture of objects in our training data the better the performance, so a trade-off must be made in terms of desired collection time. The slight dips along the linear trend, including the one at the end are likely due to the pattern at which angles were dropped out during the ablation. We would expect an more linear relationship if every number of angles were perfectly spaced apart.

Qualitative Results

In a post-study survey, we elicited qualitative responses by participants on their experience using LabelAR as well as the comparison interfaces.

Ten of twelve participants stated they would prefer to use LabelAR over the other interfaces. One found the overlayed interface easier to use, and one stated that their answer depended on LabelAR producing better results (our analysis was performed offline after the study so participants could not see the performance of models trained on their images).

Participants appreciated that they were able to place spatially stable boxes around real-world objects (four of twelve mentioned this explicitly). One commented that it was particularly useful to track multiple objects simultaneously. Suggestions for improvement largely centered around the efficiency of the widgets for initial box placement and sizing, which four participants felt could be improved. These difficulties did not prevent users from completing their tasks: *"I was able to get to a high degree of precision (regarding the boxes being drawn around the objects)*". These difficulties could be overcome with an additional round of refining the interaction design of the manipulation widgets. Also, several participants suggested dynamically changing the coloring of interface widgets based on the underlying image pixels to ensure legibility in a variety of settings.

Six of twelve participants commented positively how the capture markers provided a very engaging way to capture a variety of viewpoints: *"I really liked that it felt like a game. I loved turning the red vantage point bars green"*; *"I also liked how the red rods guided you on where to point the camera"*; *"The angle targets were fun to use"*. However, some orientations were harder to capture than others for two participants: *"I did think it was difficult to capture the rods/ handles that were nearly perpendicular to the object"*; *"In certain capturing angles (only at the top) it was sometimes difficult to line up two circles"*.

Finally, one participant noted they experienced temporary issues with the underlying tracking technology (ARKit), which in some instances lost the table plane and subsequently recomputed it several inches above or below the prior plane. As we discuss in the Limitations section, LabelAR is fundamentally dependent on the accuracy of the underlying AR tracking.

DISCUSSION

Object detection performance. We have shown LabelAR can be over 5× more precise on average than overlayed bounding box tools, and comparable or slightly better in performance to post-hoc annotation tools. This suggests that AR-based image collection tools can have a significant impact on training CV models. For this result, average precision was assessed at an IOU of 0.25, which is an acceptable evaluation criteria depending on the application. For example, in a hypothetical key-finder app, a detector does not need highly accurate boxes to indicate to a user where the keys are located. Conversely, in robotic-grasping applications, highly accurate IOU is very important since the robot needs to know exactly where the extents of the objects are to place a grabber. None of the collection apps resulted in desirable AP's at an IOU of 0.5, the best being post-hoc annotation ($AP_{0.5IOU} = 0.32$). In other words, 68% or more of the predictions made by any of the apps at this IOU-level would be false-positives. This suggests that more training images or more efficient learning algorithms are needed to have quickly-created object detectors perform well enough to create actual value for end-users in high IOU applications. Future work that can improve the bounding box accuracy of LabelAR would be highly valued.

Collection time. The collection time results from the user study show that, on average, LabelAR is over 2× faster than post-hoc annotation and competitive with the overlayed interface. The significantly larger time cost with post-hoc annotation is due to the need to label every image individually by drawing 2D boxes after the capturing process. With LabelAR, there is a relatively small upfront time cost of placing one box for each object. The time it takes to move the camera around the objects and capture the angle-markers does not depend on the number of objects and would stay constant if the number of objects were increased. The overlayed interface is the fastest method since there are no annotation tasks required by the user. However, since the user has no control over the size or ratio of the 2D box, a lot of bounding box accuracy is sacrificed for speed. It is worth noting that the time advantage of the overlayed interface over LabelAR would potentially disappear as more objects were added. The time needed to fully annotate an additional object with the overlayed interface scales with the number of images desired, whereas the time needed to annotate an additional object with LabelAR is the fixed up-front cost.

2D bounding box accuracy. LabelAR performed similarly to the overlayed interface in terms of box accuracy, but post-hoc annotation had the best accuracy over all, due to the fine-level of control the user can exercise in the Scalabel interface to place and fit nearly pixel-perfect boxes around objects in each image. Some causes of IOU degradation include poor box placement by the user (affects all three apps), fixed box ratio (overlay interface only), and projection error from 3D to 2D boxes due to spatial tracking errors (LabelAR only). Underlying AR spatial tracking errors might have significantly affected at least one of the collections in our study, where the 2D box results are all shifted down by about 40 pixels relative to the object positions. We are unable to confidently diagnose whether this was spatial tracking, user-related errors or both, so we included this data in the final results.

Diverse image perspectives. While the results show that the (free-form) post-hoc tool produces superior IOU numbers, the AP results show that LabelAR performs comparably well. We believe this is due to the diversity in image perspectives produced by LabelAR's guided capture interface. While in theory it is possible to produce a similarly diverse set of data using the post-hoc tool, our study results show that users naturally underestimate the amount of viewing angle variation needed. We show AR guided interfaces can ensure that users produce quality training sets.

Number of images taken. Our results show that while, on average, users took less images with LabelAR, detection performance is comparable or better. We attribute this to the LabelAR interface encouraging the collection of 1 - all five instances in every picture, 2 - diverse image perspectives, and 3 - partial occlusions (which create valuable 'hard examples' to learn from.)

Angle ablation. The angle ablation results inform the design choice of maximizing the number of angle-markers within the constraints of usability and time (too many angle markers might be overwhelming in terms of user experience, or take too long to capture all of them.) Future work could try to establish an upper bound on performance gains from number

of angles and find ways to collect more angles from the user without increasing time or decreasing usability.

LIMITATIONS

Some limitations of the interface we present are inherent to AR devices, while others result from assumptions that we make, such as static objects.

Large-scale collection. Our design is currently optimized for small scale tasks. Post-hoc labeling approaches have the benefit that a larger labeling task can be parallelized across multiple users, e.g. through crowdsourcing. It is conceivable to also build a crowdsourced LabelAR repository, where multiple users contributed images of similar objects. We don't investigate this scaling in our current work, but point to it as an avenue of future work.

Non-planar surfaces. Our interface can position bounding volumes anywhere in space. We found that snapping to an identified plane increases the speed of correctly positioning boxes, so this is the default behavior for the mobile application. ARKit detects planes at both horizontal and vertical orientations. For platforms that reconstruct 3D meshes we could, but don't currently, snap the bounding volumes to the mesh.

Spatial tracking reliance. The quality of the training data produced by LabelAR is influenced heavily by the AR device's ability to maintain spatial tracking of the environment. All AR devices we've tested our interface on were subject to some amount of drift, which manifests in holocubes no longer being physically on top of the real-world objects. This can cause problems leading to a decreased IOU.

Static objects assumption. An obvious limitation arises from our assumption of static objects, that is the objects must remain in-place or else the 3D bounding volumes do not track if those objects are displaced in the scene. In other words, if a wearer were to pick up and move an object of interest, the video annotation capabilities will be lost. We see this as a major limitation for machine learning research since humans tend to pick things up and manipulate them for closer examination when confronted with a novel object (especially toddlers, who are constantly engaging in interactive visual learning).

Also, our approach only works for objects of small enough size where it is easy for a user to capture different viewpoints efficiently — for example, one couldn't efficiently get a lot of different viewpoints of a large building.

Cuboid bounding volumes. Although we chose the cuboid as our bounding volume shape for easy scaling and fitting, it does not always allow for a perfect fit. In particular, there is significant space around spherical and cylindrical objects that, when projected into image space, can contribute to inaccuracies when calculating IOU.

FUTURE WORK

In addition to addressing the limitations already discussed, there are a few particularly interesting research directions we would like to pursue:

Real-time iterative model training. We are excited about possible research directions involving interactive adaptation for object recognition. LabelAR's ability to collect high quality training data in a short period unlocks a promising research direction to explore user interfaces and computer vision methods for iterative in-situ re-training that leverages user interaction. In this work, we have explored interactive collection followed by a single re-training step. In future work, we'd like to explore what the user can do with the training results and how additional in-situ re-training could benefit both the user and model.

Crowdsourcing with LabelAR. We think that LabelAR is a first step in creating a crowdsourced image database along the lines of ImageNet [10] where collections of objects annotated from a large variety of angles can be uploaded and shared. There are many significant engineering challenges in building such a system, and interface improvements would need to be built on top of LabelAR to support a hierarchical labeling structure to the data.

Extensions to robotic cameras. Our data collection interfaces need not stay limited to hand-held or head-mounted devices. Conceivably, a wearer of a VR headset could control a robotic video platform such as a drone to overcome accessibility constraints. For example, an engineer could use our system to quickly train a computer vision model to recognize particular types of cracks or visible damage on buildings or bridges that would be difficult to collect otherwise.

CONCLUSION

This paper introduced LabelAR, an augmented reality interface for collecting computer vision model training data. LabelAR utilizes the spatial tracking technology in AR devices to localize bounding volumes over physical objects in the environment, and uses these volumes to automatically label 2D bounding boxes for these objects. It also provides a guided interface to assist users in collecting a variety of viewing angles.

In a user study, we showed that LabelAR is able to collect data significantly faster than baseline post-hoc annotation tools, such as Scalabel, while producing comparable quality. We also compared LabelAR against an overlayed 2D bounding box interface, a tool designed to collect training data very quickly, to which LabelAR was able to produce significantly better results. In short, LabelAR combines the speed of a tool like the overlayed interface with the output quality of post-hoc annotation tools.

We believe our work opens up many avenues for future joint computer vision and HCI research with real-time iterative model training feedback or large scale crowdsourcing opportunities. We also believe LabelAR can serve as a new benchmark in the computer vision community as an in-situ data collection tool for training multi-object detectors.

ACKNOWLEDGEMENTS

This research was supported through the National Geospatial-Intelligence Agency (NGA) Vector program. Approved for public release under case 19-678.

REFERENCES

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. 2018. Efficient Interactive Annotation of Segmentation

Datasets with Polygon-RNN++. *CoRR* abs/1803.09693 (2018). http://arxiv.org/abs/1803.09693

[2] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2009. Overview-Based Example Selection in End-User Interactive Concept Learning. ACM Press, 247–256. https://doi.org/10.1145/1622176.1622222

[3] Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. 2018. Fluid Annotation: a human-machine collaboration interface for full image annotation. *CoRR* abs/1806.07527 (2018). http://arxiv.org/abs/1806.07527

[4] Ferran Argelaguet Sanz and Carlos Andujar. 2013. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers and Graphics* 37, 3 (May 2013), 121–136. DOI: http://dx.doi.org/10.1016/j.cag.2012.12.003

[5] Sven Bambach, David J. Crandall, Linda B. Smith, and Chen Yu. 2018. Toddler-Inspired Visual Object Learning. In *NeurIPS*.

[6] Liang-Chieh Chen, Sanja Fidler, and Raquel Urtasun. 2014. Beat the MTurkers: Automatic Image Labeling from Weak 3D Supervision. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 3198–3205.

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. *CoRR* abs/1804.02748 (2018). http://arxiv.org/abs/1804.02748

[8] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. 2007. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007), 1052–1067.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009a. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009b. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[11] Gabriel Evans, Jack Miller, Mariangely Iglesias Pena, Anastacia MacAllister, and Eliot Winer. 2017. Evaluating the Microsoft HoloLens through an augmented reality assembly application. (2017). DOI: http://dx.doi.org/10.1117/12.2262626

[12] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vision* 88, 2 (June 2010), 303–338. DOI: http://dx.doi.org/10.1007/s11263-009-0275-4

[13] Jerry Fails and Dan Olsen. 2003. A Design Tool for Camera-based Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 449–456. DOI: http://dx.doi.org/10.1145/642611.642690

[14] George W. Fitzmaurice. 1993. Situated Information Spaces and Spatially Aware Palmtop Computers. *Commun. ACM* 36, 7 (July 1993), 39–49. DOI: http://dx.doi.org/10.1145/159544.159566

[15] Daniel Göhring, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. 2014. Interactive adaptation of real-time object detectors. *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), 1282–1289.

[16] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. A Survey of Design Issues in Spatial Input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST '94)*. ACM, New York, NY, USA, 213–222. DOI: http://dx.doi.org/10.1145/192426.192501

[17] Ke Huo, Vinayak, and Karthik Ramani. 2017. Window-Shaping: 3D Design Ideation by Creating on, Borrowing from, and Looking at the Physical World. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction (TEI '17)*. ACM, New York, NY, USA, 37–45. DOI: http://dx.doi.org/10.1145/3024969.3024995

[18] PTC Inc. 2011–2018. Vuforia Engine. https://developer.vuforia.com/. (2011–2018).

[19] Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, and Kristen Grauman. 2016. Crowdsourcing in Computer Vision. *CoRR* abs/1611.02145 (2016). http://arxiv.org/abs/1611.02145

[20] Manfred Lau, Masaki Hirose, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. 2012. Situated Modeling: A Shape-stamping Interface with Tangible Primitives. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, New York, NY, USA, 275–282. DOI: http://dx.doi.org/10.1145/2148131.2148190

[21] Jürgen Leitner, Alexander Förster, and Jürgen Schmidhuber. 2014. Improving robot vision models for object detection through interaction. *2014 International Joint Conference on Neural Networks (IJCNN)* (2014), 3355–3362.

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). http://arxiv.org/abs/1405.0312

[23] Google LLC. 2013–2019. Photo Sphere. https://www.google.com/streetview/. (2013–2019).

[24] Dan Maynes-Aminzade, Terry Winograd, and Takeo Igarashi. 2007. Eyepatch: Prototyping Camera-based Interaction Through Examples. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 33–42. DOI: http://dx.doi.org/10.1145/1294211.1294219

[25] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. 2018. Learning Instance Segmentation by Interaction. In *CVPR Workshop on Benchmarks for Deep Learning in Robotic Vision*.

[26] 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 the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15)*. MIT Press, Cambridge, MA, USA, 91–99. http://dl.acm.org/citation.cfm?id=2969239.2969250

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015a. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. DOI: http://dx.doi.org/10.1007/s11263-015-0816-y

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015b. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. DOI: http://dx.doi.org/10.1007/s11263-015-0816-y

[29] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. 2008. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision* 77, 1-3 (2008), 157–173.

[30] Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. 2017. Time-Contrastive Networks: Self-Supervised Learning from Multi-View Observation. *CoRR* abs/1704.06888 (2017). http://arxiv.org/abs/1704.06888

[31] Ivan E. Sutherland. 1968. A Head-mounted Three Dimensional Display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I (AFIPS '68 (Fall, part I))*. ACM, New York, NY, USA, 757–764. DOI: http://dx.doi.org/10.1145/1476589.1476686

[32] Michael Tsang, George W. Fitzmaurice, Gordon Kurtenbach, Azam Khan, and Bill Buxton. 2002. Boom Chameleon: Simultaneous Capture of 3D Viewpoint, Voice and Gesture Annotations on a Spatially-aware Display. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 111–120. DOI: http://dx.doi.org/10.1145/571985.572001

[33] Hsiao-Yu Fish Tung, Adam W. Harley, Liang-Kang Huang, and Katerina Fragkiadaki. 2018. Reward Learning from Narrated Demonstrations. *CoRR* abs/1804.10692 (2018). http://arxiv.org/abs/1804.10692

[34] Kai Welke, Jan Issac, David Schiebener, Tamim Asfour, and Rüdiger Dillmann. 2010. Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. *2010 IEEE International Conference on Robotics and Automation* (2010), 2012–2019.

[35] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. 2015. Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer. *CoRR* abs/1511.03240 (2015). http://arxiv.org/abs/1511.03240

[36] Ka-Ping Yee. 2003. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 1–8. DOI: http://dx.doi.org/10.1145/642611.642613

[37] Tom Yeh and Trevor Darrell. 2005. Doubleshot: An Interactive User-aided Segmentation Tool. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*. ACM, New York, NY, USA, 287–289. DOI: http://dx.doi.org/10.1145/1040830.1040901

[38] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. 2018. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *CoRR* abs/1805.04687 (2018). http://arxiv.org/abs/1805.04687

[39] 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. *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), 1386–1383.