Snapper: Accelerating Bounding Box Annotation in Object Detection Tasks with Find-and-Snap Tooling

Alex C. Williams
acwio@amazon.com
AWS AI, Amazon
Santa Clara, CA, USA

Min Bai
baimin@amazon.com
AWS AI, Amazon
New York, NY, USA

Jonathan Buck
jonabuck@amazon.com
AWS AI, Amazon
Santa Clara, CA, USA

Tristan McKinney
tristamc@amazon.com
AWS AI, Amazon
New York, NY, USA

Amy Rechkemmer*
arechke@purdue.edu
Purdue University
West Lafayette, IN, USA

Koushik Kalyanaraman
koukal@amazon.com
AWS AI, Amazon
Santa Clara, CA, USA

Matthew Lease
matlease@amazon.com
AWS AI, Amazon
Santa Clara, CA, USA

Patrick Haffner
haffnerp@amazon.com
AWS AI, Amazon
New York, NY, USA

Xiong Zhou
xiongzhou@amazon.com
AWS AI, Amazon
Seattle, WA, USA

Kumar Chellapilla
chelkuma@amazon.com
AWS AI, Amazon
Santa Clara, CA, USA

Li Erran Li
lilimam@amazon.com
AWS AI, Amazon
Santa Clara, CA, USA

ABSTRACT

Object detection tasks are central to the development of datasets and algorithms in computer vision and machine learning. Despite its centrality, object detection remains tedious and time-consuming due to the inherent interactions that are often associated with drawing precise annotations. In this paper, we introduce Snapper, an interactive and intelligent annotation tool that intercepts bounding box annotations as they're drawn and "snaps" them to the nearby object edges in real-time. Through a mixed-design user study with 18 full-time annotators, we compare Snapper's annotation mode to alternative modes of annotation and find that Snapper enables participants to complete object detection tasks 39% more quickly without diminishing annotation quality. Further, we find that participants perceive Snapper as a tool that is interactively intuitive, trustworthy, and helpful. We conclude by discussing the implications of our findings as they relate to augmenting annotators' conventions for drawing annotations in practice.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; *User studies*; *Graphical user interfaces*; *Interactive systems and tools*; • **Computing methodologies** → *Object detection.*

KEYWORDS

Assisted annotation, object detection, annotator productivity.

*Work completed during an internship at Amazon.

IUI '24, March 18–21, 2024, Greenville, SC, USA
2024. ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Alex C. Williams, Min Bai, Jonathan Buck, Tristan McKinney, Amy Rechkemmer, Koushik Kalyanaraman, Matthew Lease, Patrick Haffner, Xiong Zhou, Kumar Chellapilla, and Li Erran Li. 2024. Snapper: Accelerating Bounding Box Annotation in Object Detection Tasks with Find-and-Snap Tooling. In *29th ACM Conference on Intelligent User Interfaces, March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Data annotation is a task that is central to modern artificial intelligence and machine learning. Historically, data annotation and labeling have been facilitated by public-facing platforms, such as Amazon Mechanical Turk and Prolific, that engage the general public in completing various types of tasks in exchange for compensation [40, 85]. Alongside these public-facing platforms, data labeling and annotation efforts have continued to become commonplace in private industry firms who formally employ thousands of individuals as full-time annotators [37]. Signifying the societal relevance of these firms, market estimates for third-party data labeling solutions providers are projected to grow upward of 4.1 billion USD by 2024 [18]. Similarly, governments are partnering with these annotation firms to bring data labeling jobs to more remote parts of their countries [88]. The growing prominence of data annotation highlights the importance of research that aims to improve annotators' productivity through innovation and design.

An important facet of modern annotation is that it remains significantly tedious. Among the most common modes of annotation for object detection in 2D images, bounding box annotation asks annotators to create rectangular annotations that bound objects of relevance as tightly as possible [2]. In contrast to the simplistic nature of bounding box annotations, semantic segmentation asks

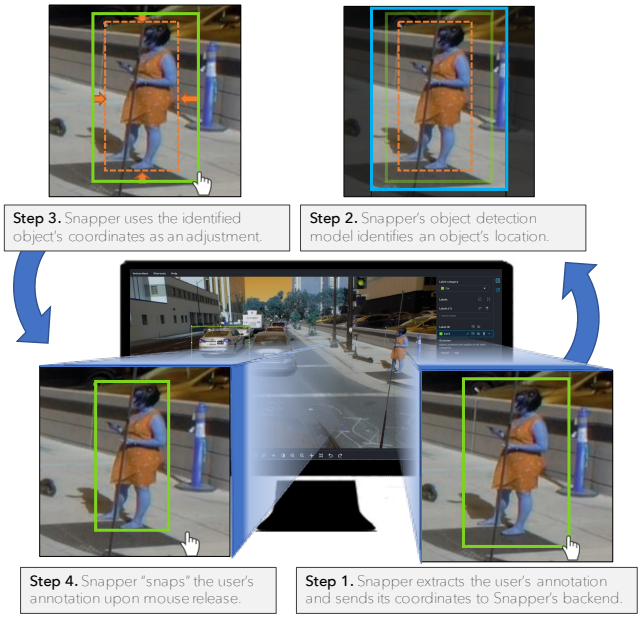


Figure 1: Snapper enables annotators to be more productive by automatically tightening their annotations with "snappable", model-generated adjustments in real-time.

annotators to create fine-grained, free-form annotations that annotate an object's boundary with point-level annotations, requiring an order of magnitude more precision [53]. Prior research has aimed to reduce the time-consuming nature by introducing new techniques for accelerating the annotation process (e.g., minimizing the number of mouse clicks for an annotation [66]). However, the majority of these proposed tools have been evaluated by means of simulation, leaving much to be learned about their practical efficacy.

In this paper, we explore how aspects of annotator productivity can be augmented with intelligent tooling in image annotation contexts. We introduce and study *Snapper*, an interactive and intelligent system that automatically adjusts bounding box annotations in real-time. Snapper facilitates adjustments by intercepting and forwarding all direct manipulation events on the annotation canvas (e.g., mousedown, mousedrag, and mouseup) through a set of secondary event handlers that forward annotation coordinate information to the system's bounding box adjustment model. As shown in **Figure 1**, Snapper's user interface receives annotation updates to adjust an annotator's annotation immediately upon mouse release. In a user study with 18 annotators, we find Snapper's mode of annotation to be significantly usable, intuitive, and generally helpful. We also find that it enables annotators to complete annotation tasks more quickly with no detriment to label quality in a way that existing techniques do not. In this paper, we specifically:

- Introduce *Snapper*, an interactive and intelligent system that automatically adjusts bounding box annotations in real-time.
- Examine how Snapper supports image annotation practices in comparison to conventional bounding box annotation and the state-of-the-art Extreme Clicking annotation mode [66].
- Find that Snapper reduces total task time by 39% and cumulative annotation creation and edit time by 95% without diminishing the quality of annotations for objects that are not small.

The remainder of this paper is structured as follows. We first provide an overview of related work at the intersection of human-computer interaction and machine learning. We then present Snapper alongside the components that facilitate its auto-adjustment utility. We present our approach for evaluation and conclude with a discussion of considerations for designing new interactive and intelligent tools that empower annotators in unique and powerful ways.

2 RELATED WORK

In this section, we provide an overview of the work related to assistive annotation tools, interface-supporting snapping, and interface intelligence.

2.1 Snapping as an Assistive Technique

Snapping [10, 11, 79] is a common interface technique that allows graphical objects to "snap" to "snap locations" within a given graphics space via direct manipulation. In contrast to manually positioning graphical objects, snapping supports object positioning by facilitating exceptional precision with minimal effort [7]. Importantly, traditional snapping hinges on the availability of "snap locations". The concept of snapping originates from Sutherland's *Sketchpad* that allowed users to draw graphical lines between two points on a grid that lines would snap to [79]. Bier and Stone introduced "snap-dragging", a cursor-based technique that allows graphical objects in 2D space to snap together during drag operations using the presence of other graphical elements rather than a pre-existing point grid [11]. Later, Bier demonstrated how the snap-dragging technique can be used in three-dimensional space [10]. Today, snapping exists as a standard feature in many applications as observed by its presence in Microsoft PowerPoint, Google Slides, SketchUp, AutoCAD, and more. Research has continued to demonstrate how snapping can support new interactive experiences, ranging from accessing remote screen content [6] to improving precision in the construction of fabricated objects [62].

Prior research has demonstrated how snapping can be applied to settings where "snap locations" are unknown. For example, Gleicher introduced and demonstrated *"image snapping"*, a technique that uses image features (e.g., shape origins, shape edges, or pixels with specific colors) to identify areas of an image that an interface can "snap" to [24]. Similar approaches have been adapted to facilitating snapping in the augmented reality uses-cases, such as drawing fine-grained annotations on mobile devices [47, 48] or determining alignments between physical and virtual objects [64]. Inspired by Bier [10], Szalavári et al. [80] introduced *"face-snapping"*, a technique for snapping objects in virtual reality in which shapes snap together like puzzle pieces based on constraints of the virtual object's physical shape. More recent research on snapping has focused on "smart interaction techniques" in which elements of traditional snapping are augmented by design. One such example is Baudisch et al. [7]'s *Snap-and-Go* technique, which facilitates snapping in 2D and 3D without a deactivation function by augmenting the motor space at the snap location.

2.2 Techniques for Assisted Annotation

2.2.1 Drawing and Image Editing Applications. Researchers have explored a variety of assistive tools and techniques for improving annotation within drawing and image editing software applications. The goal of these techniques is often centered around reducing the time and cognitive effort required to annotate an object in an image (e.g., with a selection) [12]. Adobe Photoshop's *Magic Wand* is one such tool that assists users in selecting a part of the image with a specified configuration of tolerance and anti-aliasing. Practical studies of the tool have repeatedly demonstrated its utility for accelerating the task of object selection in a variety of image contexts [75, 89]. Similarly, Adobe Photoshop's *Magnetic Lasso* allows users to identify boundaries by providing a rough, contoured outline around an object boundary, a technique that was first seen in Mortensen and Barrett [61]'s *"Intelligent Scissors"* concept. Device-specific tools have facilitated "sloppy selection", such as Lank and Saund [46]'s work, that allows object selection to be inferred via an analysis of motion dynamics of an input device rather than the literal stroke that it emits. Further extensions have focused explicitly on assistive selection for sketch segmentation and selection [63, 86]. Minimizing the number of modes in these contexts is an important issue [76].

Such assistive tools are generally fueled by procedures that strategically manipulate or query the underlying graphics space. Chuang et al. [17] introduced a Bayesian approach for separating foreground and background objects with "trimaps". Inspired by this, Boykov and Jolly [13] studied *Graph Cut*, an optimization technique that achieves robust object segmentation when color distributions between foreground and background are not well separated. Rother et al. [71] introduced *GrabCut*, an iterative version of the Graph Cut technique. Most relevant to our work, Li et al. [49] introduced *lazy snapping*, a graph-based computer vision technique that aims to assist users in automatically segmenting objects. The interaction design facilitated by the technique requires that users provide an initial contour of a boundary that the technique will automatically "snap" to a nearby edge. In a comparison with Adobe Photoshop's *Magnetic Lasso*, Li et al. observed that lazy snapping reduced the participants' use of the undo tool by 20%, reduced participants' drawing time by 60%, and was generally described as "much easier" and "almost magic".

2.2.2 Crowdsourced Annotation. Image annotation is a type of crowdsourced task that asks crowdworkers to create graphical annotations that either segment an object (e.g., with a fine-grained set of free-form points) or bound an object (e.g., with a bounding-box) [36]. It is generally well-understood that the task of creating and modifying either type of annotation is time-consuming and tedious because precision is paramount [66]. In a 2014 survey deployed to CrowdFlower, Gadiraju et al. [23] reported that annotation-related tasks (i.e., identified as `VV` tasks and `IA` tasks) account for half of the tasks deployed to the platform. As machine learning practice and research has further swelled in recent years and continues to rely heavily on crowdsourced annotation to generate large datasets (e.g., [43]), such annotation tasks have only increased in frequency, scale, and importance.

A growing body of research has explored new techniques for minimizing annotation effort. For example, one line of work asks annotators to identify points that "matter" in determining the boundary of a particular object [41, 55, 87]. Conceptually, these approaches draw from Papadopoulos et al. [66] who introduced *"extreme clicking"*, a technique in which annotators label an object with a finite set of points on the 2D image that bound an object. Papadopoulos et al. report a comparative study that indicates "extreme clicking" allows annotators to label objects, on average, in 7 seconds per box without diminishing annotation quality. Maninis et al. [55] introduced *DEXTR*, a technique that operationalizes the "extreme clicking" method. Benenson et al. [9] further demonstrated the viability of the technique's utility to produce high-quality labels by using it to produce masks for 2.5 million instances in the OpenImages dataset. In a study with workers on Amazon Mechanical Turk, Bearman et al. [8] demonstrated that asking annotators to issue only a single click at the center of an object is a viable alternative to issuing several clicks at the boundary of an object. A similar approach was also explored by Papadopoulos et al. [67]. Subsequent research has explored a myriad of individuals extensions aimed at improving the state of interactive object segmentation (e.g., by allowing corrections to machine predictions) [3, 42, 51]. Despite being recognized as state-of-the-art technology, the vast majority of these tools and techniques have been evaluated only through simulation, with little known about their efficacy in practice.

2.3 Contribution

In summary, prior work has demonstrated the value in exploring new techniques to reduce annotation effort. In this paper, we leverage the successes of prior work to motivate the design of *Snapper*, a new interactive system that leverages facets of interface snapping to reduce the laborious nature of drawing bounding box annotations by hand. Alongside Snapper, we present findings from a user study with full-time data annotators that compares the system's performance to state-of-the-art techniques that assist users in drawing annotations by hand (i.e., extreme clicking).

3 SNAPPER

In this section, we introduce *Snapper*, an interactive and intelligent system "snaps" ill-fitted object annotations to image-based objects in real-time. We begin by discussing the concept behind the Snapper system and propose a design space for machine-generated recommendations for adjusting human annotations in real-time. We then discuss Snapper's system architecture.

3.1 Conceptual Motivation

Object annotation is a time-consuming and tedious task that requires annotators to create annotations that "tightly" fit an object's boundaries. Bounding box annotation tasks, for example, require annotators to ensure that all edges of an annotated object are enclosed in the annotation. Furthermore, object segmentation tasks require annotators to bound all edges of an object with a free-form annotation. In practice, creating annotations that are precise and well-aligned to object edges is laborious and fatiguing [1].

We argue that "noisy" labels are a by-product of the annotation tools that exist today. We hypothesize that HCI research can

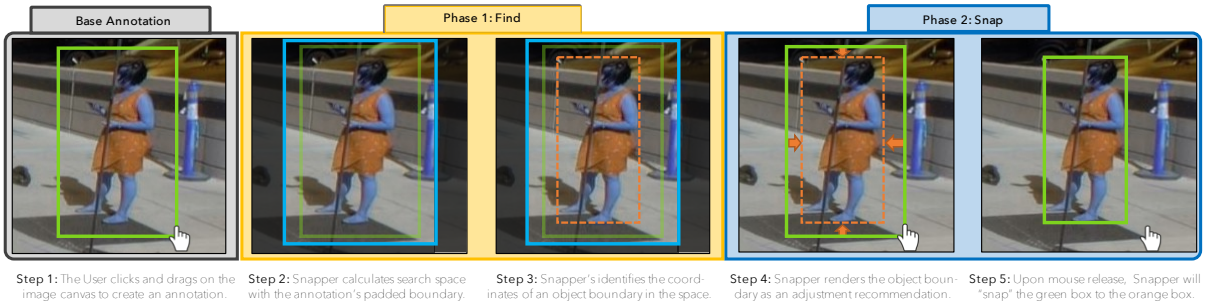


Figure 2: Snapper's Find-and-Snap procedure for generating and rendering annotation adjustment recommendations.

embrace the "noise" in human annotation and explore both new and existing techniques alike in support of "de-noising" annotation data at the time of collection. In particular, we investigate how a well-known technique – *snapping* – can translate to the domain of crowdsourced object annotation. In support of this goal, we designed *Snapper* as an interactive tool that automatically adjusts annotators' noisy annotations, allowing them to accelerate their work and spend less time being tedious in their labeling.

3.2 System Architecture

Snapper is an interactive and intelligent system that automatically"snaps" object annotations to image-based objects in real-time. The system's implemented is entirely web-based and is composed of two sub-systems. The first sub-system is a front-end ReactJS component that intercepts annotation-related mouse events and handles the rendering of recommendation information. The system's front-end architecturally complements the majority of web-based annotation interfaces by relying only on the HTML <canvas> element in conjunction with JavaScript event handlers. This design principle enables other crowdsourced annotation platforms (e.g., Zooniverse [77]) to adopt or replicate Snapper-like interfaces as described here. The second sub-system is a web server receives requests from the front-end client, routes the requests to an machine learning model to generate adjusted bounding box coordinates, and sends the data back to the client. For our study, we implemented and deployed Snapper within AWS SageMaker Ground Truth[1], a large-scale, commercial platform for data labeling and annotation.

3.3 The Find-and-Snap Technique

Inspired by design guidelines at the intersection of HCI and AI [4, 35], Snapper's interaction design centers around the *Find-and-Snap* technique, a model-based interaction technique that identifies objects in images via localization and "snaps" to the identified object locations. The Find-and-Snap technique conceptually draws upon mouse release. Shown in **Figure** 2, the Find-and-Snap technique consists of two phases. First, *Phase 1: Find* (Section 3.3.1) localizes a candidate object an image based on an initial human annotation. Next, *Phase 2: Snap* (Section 3.3.2) can asynchronously "snap" the user's current annotation to an adjustment generated by the system.

[1]https://aws.amazon.com/sagemaker/groundtruth/

3.3.1 Phase 1: Find. Upon invocation, the Find-and-Snap technique initiates the process of localizing an object by sending two parameters to Snapper's bounding box adjustment model:

(1) *Annotation Coordinates [List]*: The four coordinates of the user's initial bounding box annotation.
(2) *Base Image URL [String]*: The URL of the image from which the coordinates are associated.

In the context of our deployed version of Snapper, Snapper's front-end implementation extracts each of these parameters automatically from the annotation platform's user interface in which it is implemented.

Conventional bounding box annotation typically involves interaction designs in which annotations are created by click-and-drag operations by which the mouse-up and mousedown events correspond to creation and deletion. When drawing annotations with this interaction design, annotators may be significantly imprecise in their initial positioning of the annotation being drawn. We therefore implemented a third, optional parameter named *Buffer Ratio* that Snapper will use to scale the user's initial annotation coordinates by the associated image's height and width before being sent to the model for adjustment (See Figure 2; Step 2). By default, the Buffer Ratio parameter is set to 1.0. A larger Buffer Ratio will increase the scale of the coordinates while a smaller value will do the opposite. The Find phase concludes after having sent all parameters to Snapper's adjustment model and received a response with a valid set of four coordinates that represent the updated location of the initial bounding box.

Toward the goal of studying the effect of Snapper's adjustment accuracy, we implemented a configurable setting that allows Snapper to control where object localization requests in the *Find* phase are routed:

(1) Dynamic Model (DM): The *Find* technique adjusts user's initial annotations using Snapper's adjustment model.
(2) Ground Truth (GT): The *Find* technique adjusts user's initial annotations using ground truth annotation data.

3.3.2 Phase 2: Snap. In its *Snap* phase, Snapper's primary goal is two-fold: (1) preparing proposed adjustments and (2) "snapping" the user's annotation to the recommendation. Upon releasing the mouse, Snapper will adjust the user's annotation coordinates to the recommendation with an interpolated visual that mirrors the popular "snap-to-grid" metaphor [7]. After the Snap is complete, the adjusted annotation can be iteratively edited to the user's liking.

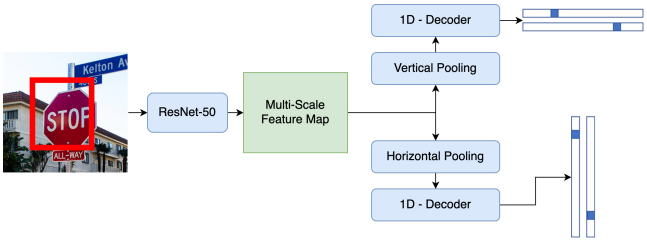


Figure 3: Snapper Bounding Box adjustment Model Design.

3.4 A Model for Bounding Box Adjustments

A tremendous number of high-performing object detection models have been proposed by the computer vision community in recent years [15, 52, 69, 70]. However, these state-of-the-art models are typically optimized for *unguided* object detection. In support of facilitating Snapper's "snapping" functionality for adjusting users' annotations, the input to our model is an initial bounding box, provided by the annotator, which can serve as a marker for the presence of an object while the output space is a single bounding box. Furthermore, as the system has no intended object class it aims to support, Snapper's adjustment model should be object-agnostic such that the system performs well on a range of object classes. In general, these requirements diverges substantially from the use-cases of these prior models.

3.4.1 Model Architecture. To enable Snapper with the ability to adjust users' annotations, we design and implement a machine learning model for *bounding box adjustment* as shown in **Figure 3**. As input, the model takes an image and a corresponding bounding box annotation. The model extracts features from the image using a convolutional neural network based on ResNet-50 [30]. Following feature extraction, directional spatial pooling is applied to each dimension to aggregate the information needed to identify an appropriate edge location. As output, the model returns the four final classification vectors identified in the previous step. The model was implemented in PyTorch.

3.4.2 Training Data: A Dataset of Noisy Object Annotations. Snapper's end-goal is to improve the accuracy of noisy user annotations for arbitrary class objects by generating and applying annotation adjustments. As collecting human annotation data was cost-prohibitive, we employ an approach in which we add noise to a publicly available dataset of image object annotations. In support of our goal for object-agnostic adjustments, we source our data from the MS COCO dataset [50], which contains 1.5 million object annotations across 91 classes in 330k images. Using an official train, validation, and test split of the MS COCO dataset, we dynamically generate noisy annotation data by randomly adjusting the ground truth bounding box coordinates with "jitter". Our procedure for adding "jitter" first shifts the center of the bounding box by up to 10% of the corresponding bounding box dimension on each axis and then rescales the dimensions of the bounding box by a randomly sampled ratio between 0.9 and 1.1. The jittering procedure was applied to adjust the positioning of 860,000 ground truth annotations. The product of this procedure is visualized in **Figure 4**.

Figure 4: Examples of noisy bounding boxes generated by the jittering procedure used to prepare training data for Snapper's adjustment model. Unaltered ground truth boxes are shown in green while jittered boxes are in shown in red.

We train our model using all 80 object types in the MS COCO dataset covering a large variety of classes. Furthermore, the specific semantic class information is not passed to the model. Because of these design choices and the dataset's tremendous size, we find that our trained model is able to generalize not only to the breadth of object classes within the MS COCO dataset, but also to other common datasets with a generic set of object classes (e.g., PASCAL VOC 2012 [21]).

3.4.3 Model Evaluation. We evaluated Snapper's adjustment model using a type of evaluation standard to object detection models that employs two measures to examine validity: Intersection over Union (IoU), Edge Deviance, and Corner Deviance [81]. Commonly used to assess quality in standard object detection tasks, IoU calculates the alignment between two annotations by dividing the annotations' area of overlap by the annotations' area of union, yielding a metric that ranges from 0 to 1. However, as our system is aimed at generating bounding boxes of annotation quality, we note that the edges of a relatively large bounding box with high IoU may nonetheless be insufficiently accurate at the pixel level. This motivates the addition of the Edge Deviance and Corner Deviance metrics, which are calculated by taking the fraction of edges and corners that deviate from the ground truth by a pixel distance. Here, we apply these metrics to the validation set from the official MS COCO used for training. We specifically calculate the fraction of bounding boxes with IoU exceeding 90% alongside the fraction of Edge Deviations and Corner Deviations that deviate less than 1 or 3 pixels from the corresponding ground truth.

Furthermore, we examine the ability of Snapper to produce bounding box edges with higher degrees of precision as compared with models trained using the traditional object detection objectives. To this end, we apply Snapper to the detection output of DETR [15] (a current transformer-based state-of-the-art object detector) used as "noisy" input data and compare the quality of the refined detection results to the input.

Table 1: Performance of Snapper's bounding box adjustment model when applied to refine annotations in two different sources of noisy data: Jittered MS COCO and DETR.

	IoU		Edge Deviation		Corner Deviation	
Source	**mIoU**	**> 90%**	**< 1px**	**< 3px**	**< 1px**	**< 3px**
COCO [50]	81.9%	8.6%	30.4%	57.9%	12.0%	38.1%
COCO [50] + Snapper	89.0%	51.9%	52.6%	80.4%	30.1%	65.5%
DETR [15]	76.2%	33.1%	34.1%	66.6%	12.3%	45.7%
DETR [15] + Snapper	76.6%	34.9%	40.5%	68.1%	18.2%	48.8%

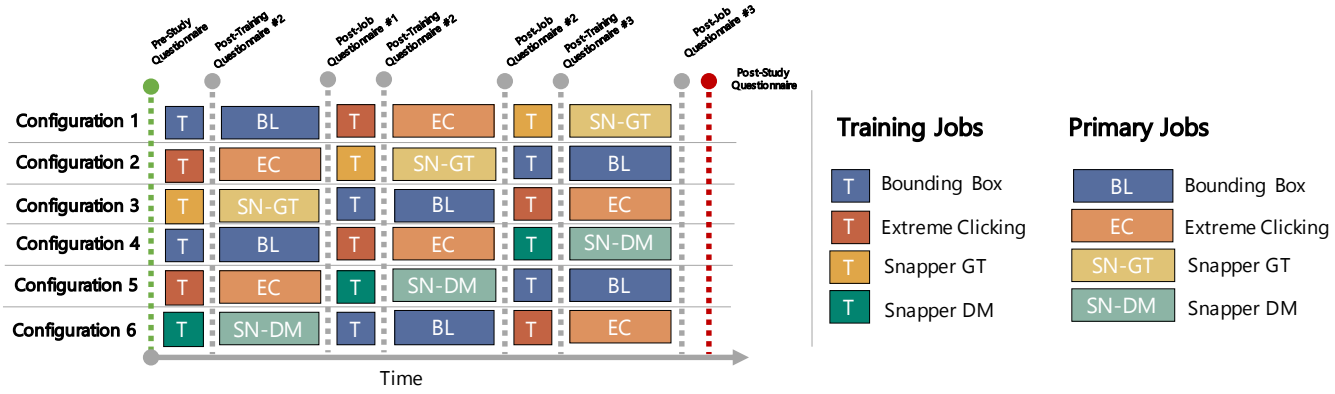


Figure 5: Our counterbalanced, mixed-design study with six possible configurations with four condition types: Baseline (BL), Extreme Clicking (EC), and Snapper with Ground Truth (SN-GT), and Snapper with Dynamic Model (SN-DM).

As shown in Table 1, the output of Snapper's adjustment model is improved over the two sources of noisy data across each of the three metrics. The improvements are especially notable in the pixel-level precision metrics, which are crucial requirements for the acceptability of annotation quality labels. We observe that applying Snapper to the Jittered MS COCO dataset improved IoU by upward of 40%. While the quality gain from using Snapper over the DETR detection output is lower, we still note that Snapper is able to increase the fraction of edges and corners within 1px of the ground truth by more 19% and 48%, respectively. These findings support our decision to create and use a custom, task-specific model design.

4 USER STUDY

Snapper seeks to accelerate image annotation tasks without sacrificing annotation quality. In support of this goal, we designed a "first-use" study [29] to understand the strengths and shortcomings of the system.

4.1 Research Questions

We motivate our study with Snapper using three research questions:

RQ1. How does Snapper support annotation practices for image object annotation?
RQ2. How does annotators' use of Snapper affect annotation quality?
RQ3. How does annotators' use of Snapper affect annotation time?

Unlike prior approaches [66, 72], we employ a mixed-methods approach to evaluate our tools through the combined lens of qualitative and quantitative data. To address **RQ1**, we ask open-ended questions related to Snapper's impact and administer validated instruments to assess system usability, team performance, and perceived cognitive load. To address **RQ2** and **RQ3** respectively, we collect image annotations on images for a ground-truth dataset alongside time-stamped telemetry data that details activities that take place in the annotation interface.

4.2 Study Design

To address our research questions, we developed a mixed design study that includes elements of within-subjects and between-subjects studies as shown in **Figure 5**. Through the study's between-subjects design, we compare annotators' use of Snapper with a baseline mode of annotation – conventional bounding box annotation – and a state-of-the-art mode of annotation that prior research has identified as being notably fast – Extreme Point annotation [66]. The study conditions were defined as follows:

- **Condition 1: Baseline: Traditional Bounding Box (BL)**. Participants manually annotate all image objects via standard bounding boxes, following the standard interaction design for 2D image annotation tasks.
- **Condition 2: Extreme Clicking (EC)**. Participants manually annotate all image objects via Papadopoulos et al. [66]'s four-point "extreme clicking" technique, yielding a bounding box upon the fourth click.
- **Condition 3: Snapper (SN)**. Participants manually annotate all image objects via standard bounding boxes with the availability of Snapper's "Find-and-Snap" tool. In support of addressing all three **RQs**, we explore two versions of Snapper that distinctly source their adjustment recommendations from Snapper's bounding box adjustment model or from direct comparisons to ground truth data:
  - **Condition 3A: Snapper - Ground Truth (SN-GT)**. Snapper's model backend will generate annotation adjustment recommendations for relevant objects statically using known ground truth information.
  - **Condition 3B: Snapper - Dynamic Model (SN-DM)**. Snapper's model backend will generate annotation adjustment recommendations for objects dynamically using its object detection model.

Our study design's within-subjects nature requires each participant to engage in all three conditions. The within-subjects design seeks both to eliminate the effect of individual differences [26] and the unavailability of an appropriate baseline measure of performance

for traditional bounding box annotation tasks and new state-of-the-art techniques [66, 67]. In contrast to the within-subjects design, the between-subjects nature of our study requires participants to engage with only one version of the Snapper tool. The motivation behind this study design decision is driven by the possibility of a learning effect that would arise in participating in both Snapper conditions (e.g., biases about accuracy developed in 3A would likely affect participants' interactions with the tool in 3B). In circumstances where such effects are suspected, experimental design advises a between-subjects design.

4.3 Task Design: Object Detection

Prior studies of new interaction techniques for 2D annotation frequently utilize image dataset benchmarks, such as DAVIS17 [38, 65], Cityscapes [51], PASCAL VOC 2007 [66, 67], PASCAL VOC 2012 [8, 42, 66] and MS COCO [3, 9, 42, 67], to evaluate their efficacy. A caveat of these studies is that they are often evaluated through simulation because they are predominantly published in computer vision and machine learning venues where user studies may undergo less scrutiny [3, 9, 42, 51, 65, 67]. Further, the studies that involve human subjects often fail to report significant detail (e.g., the number of recruited annotators [8]). The complications are further exaggerated by divergences in task design, such as the number of tasks that annotator complete (e.g., 10 images [66] vs. 20 images [67]) or the number of object classes to be labeled (e.g., 1 object class [66] vs. 65 object classes [9]).

We designed an object detection task for 2D images that, to the best of its ability, was inspired by task designs in prior research. We used the PASCAL VOC 2012 dataset's images and associated annotations [21] as it remains one of the most widely used dataset benchmarks in computer vision and machine learning research, specifically for evaluating interactive annotation techniques. Additionally, From the dataset's 11,530 images, we randomly sampled 90 images. The sampling criteria required that an image must contain at least two known instances of relevant objects classes that require annotation. From the dataset's 20 object classes, we ask participants to identify one object class from each of the dataset's four types of object class types (i.e., Person, Animal, Vehicle, Indoor). The object classes are: *Person*, *Sheep*, *Bicycle*, and *Chair*. Alongside class presence, our sampling criteria required that half of the 90 images have at least one object instance that was flagged as "occluded" in order to ensure that our subset of images reflected the reality that up to 70% of a single object classes' instances can be occluded [20]. Finally, we required that each image have between three to five objects.

We divided the subset of 90 sampled images into three set of 30 images (i.e., Block 1, Block 2, and Block 3). Each block was split evenly between 15 images with no occluded objects and 15 images with at least one occluded object. Blocks 1, 2, and 3 were respectively assigned to each participant's first, second, and third condition. Task queues within each block were randomized for each participant to prevent any statistical effects related to image ordering.

4.3.1 Measuring Task Accuracy. Mirroring the evaluation described in Section 3.4.3, we use our Intersection over Union (IoU) as our primary measure of annotation accuracy following a wealth of prior work [32, 45]. For any ground truth annotation that does not have a corresponding user-generated annotation (i.e., a false negative), an IoU value of 0 is assigned to the bounding box. We map user-generated annotations to the ground-truth data on the basis of object class and IoU via the Hungarian algorithm [44].

4.3.2 Measuring Object Size. Alongside annotation accuracy, we employ a procedure popularized by Hoiem et al. [34] for applying a "size" label to all ground truth annotations used in our dataset. We calculate the area of each ground truth annotation and then bin the object into one of five possible "size" labels that are calculated relative to the object sizes for the given object class. The five labels with their distinctions are as follows with respect to the given object's class: *Extra Small* (between 0 and 10 percentile of object size), *Small* (between 10 and 30 percentile of object size), *Medium* (between 30 and 70 percentile of object size), *Large* (between 70 and 90 percentile of object size), and *Extra Large* (between 90 and 100 percentile of object size). When a user annotation is matched to a ground truth annotation for the accuracy calculation described in Section 4.3.1, the ground truth's size label is mutually applied to the user's annotation data.

4.4 Data Collection

We collected the following data as a part of the study:

4.4.1 Pre-Study Questionnaire. We inquired about participants' prior experience with bounding box annotation task and prior experience with the commercial annotation interface.

4.4.2 User-Generated Annotation Data. Across all conditions, we collected annotation data that was generated both by the user and by the Snapper system. In addition to collecting the "final" annotation data that was submitted by participants, we collected data about the "Visual Recommendation" annotations produced by Snapper and the corresponding user-generated annotation that existed on the annotation canvas.

4.4.3 Interface Telemetry Data. We collected high-level interface telemetry data (i.e., activity logs) about participants' interface activities [19, 33, 74]. We limit our telemetry analysis to two events: (1) annotation creation and (2) annotation edit. For bounding box annotation, an annotation create event begins when a "mousedown" event occurs on the annotation canvas and ends when a consecutive "mouseup" event takes place. For extreme point annotation, an annotation create event begins when the first extreme-point is issued to the annotation canvas via a "mousedown" event and ends when the fourth consecutive "mousedown" event takes place. Edit events for both bounding box annotation and extreme point annotation occur under identical circumstances (e.g., a "mousedown" event followed by consecutive "mouseup" event that updates the position of an annotation). All events were denoted with a client-side timestamp.

4.4.4 Post-Training Questionnaire. Following the completion of each training job, we administered a questionnaire that asked participants to state their agreement with the following statement: *"I understand how to make annotations with the annotation tool used in the training job that I just completed."* followed by a free-form text field that allowed them to provide additional information if any misunderstanding was present.

Table 2: Post-Questionnaire statements for SN-GT and SN-DM conditions. Agreement and Importance were reported on a standard Likert scale from 1 to 5. For Agreement, 1 indicated "Strongly Disagree" and 5 indicated "Strongly Agree".

#	Statement	Measure	Scale
1	*I trust the Find-and-Snap tool.*	Agreement	Likert (1-5)
2	*Working with the Find-and-Snap tool improved my efficiency.*	Agreement	Likert (1-5)
3	*Working with the Find-and-Snap tool improved my quality of work.*	Agreement	Likert (1-5)
4	*The Find-and-Snap tool increased the productivity of the team.*	Agreement	Likert (1-5)
5	*The Find-and-Snap tool was necessary to successfully complete the task.*	Agreement	Likert (1-5)
6	*I was necessary to the successful completion of the task.*	Agreement	Likert (1-5)
7	*If given a choice, I would work with Find-and-Snap tool again.*	Agreement	Likert (1-5)

4.4.5 Post-Condition Questionnaire. Following the completion of each condition, we administered a questionnaire on the basis of the condition being completed. A total of three post-condition questionnaires were completed by each participant as shown in Figure 5.

- **All Conditions**. Following the completion of the each condition, we administered a questionnaire that included questions from the NASA-TLX instrument to measure cognitive load [28] and the System Usability Scale (SUS) to measure system usability [5].
- **SN-GT and SN-DM Conditions**. In addition to the NASA-TLX and SUS instruments, the Post Condition Questionnaire for the Snapper conditions administered the set of statements in Table 2. This includes six statements of the Team Performance and Productivity instrument [25] and three statements related to annotation quality and annotation time. Inspired by established principles of human-AI interaction [4, 35, 59], the questionnaire concluded by asking participants to state the importance of the Find-and-Snap tool's capabilities that span four relevant dimensions: accuracy, speed, affordance, and explainability.

4.4.6 Post-Study Questionnaire. We concluded the study by administering a post-questionnaire that included three questions. The first two questions asked participants to indicate which of the three approaches that they prefer as it relates to (1) being accurate in their labeling and (2) being fast in their labeling. Participants were then asked to rank their preference of the annotation experiences explored across the three conditions.

4.5 Methods of Analysis

To test for differences in ordinal or continuous data collected between Snapper conditions, we employ Mann-Whitney U-tests [56], a non-parametric test for examining differences between two unpaired groups. To test for differences in ordinal or continuous data collected for each participant across the three conditions, we employ Freidman tests [22], a non-parametric test similar to the Kruskal-Wallis test and designed explicitly for repeated measures settings. We employ Thematic Analysis [14] to identify themes in responses to open-ended questions in the Post-Condition Questionnaires.

4.6 Participants

"Annotators" have been historically present in private workforces (e.g., the Linguistic Data Consortium [54]) as well as those that work through public-facing crowdsourcing platforms, such as Amazon Mechanical Turk, Prolific, or Crowdworkers. Recent studies with human subjects have gravitated toward using the latter to examine the feasibility of new annotation techniques or procedures [8, 9, 66, 67]. Over the past decade, large technology corporations have created similar platforms that are intended for internal use only, such as Microsoft's UHRS [84]. The widespread proliferation of annotation and its role in machine learning has lead to the development of "annotation workforces" that engage in annotation tasks as full-time employees and often have access to a readily-available queue of annotation tasks [60, 83]. Understanding the individual differences between these populations remains an on-going effort within CSCW and beyond it [58, 78].

We recruited participants from an annotation workforce at a large technology corporation in which annotators are hired as full-time employees. Our decision is motivated by several factors. First, annotation workforces remain largely unstudied as a population despite growing increasingly more common throughout the private technology sector. Second, annotation workforces are, in many ways, the ideal population to study the efficacy of new interaction techniques as their attention is focused entirely on completing annotation tasks that exist within their queue of work. In contrast, research has demonstrated that other populations of annotators can be subjected to scenarios of divided attention as a by-product of economic incentives (e.g., workers on Amazon Mechanical Turk multitasking across other HITs or finding additional work [82, 85]). Conducting our study with an annotation workforce enables us to speak more concretely about the ecological validity of Snapper's efficacy.

4.6.1 Recruitment Methodology. We used snowball sampling via an internal mailing list to recruit participants for our study. Our initial email extended an offer of study participation to full-time annotators and included a description of our study which aimed to "better understand how a new interactive tool affects annotator productivity". As annotators are accustomed to drawing from a task queue, we framed study participation as "a standard annotation task" that would be added to their work queue upon agreeing to participate in order to reduce biases related to study awareness.

4.6.2 Study Procedure. Following recruitment, each participant was randomly assigned to one of the six configurations detailed in **Figure 5** and then deployed a total of three primary labeling jobs (i.e., via *Anonymous Platform*) corresponding to the three sub-conditions tied to their study configuration. Three training jobs were also launched for each condition to provide annotators with

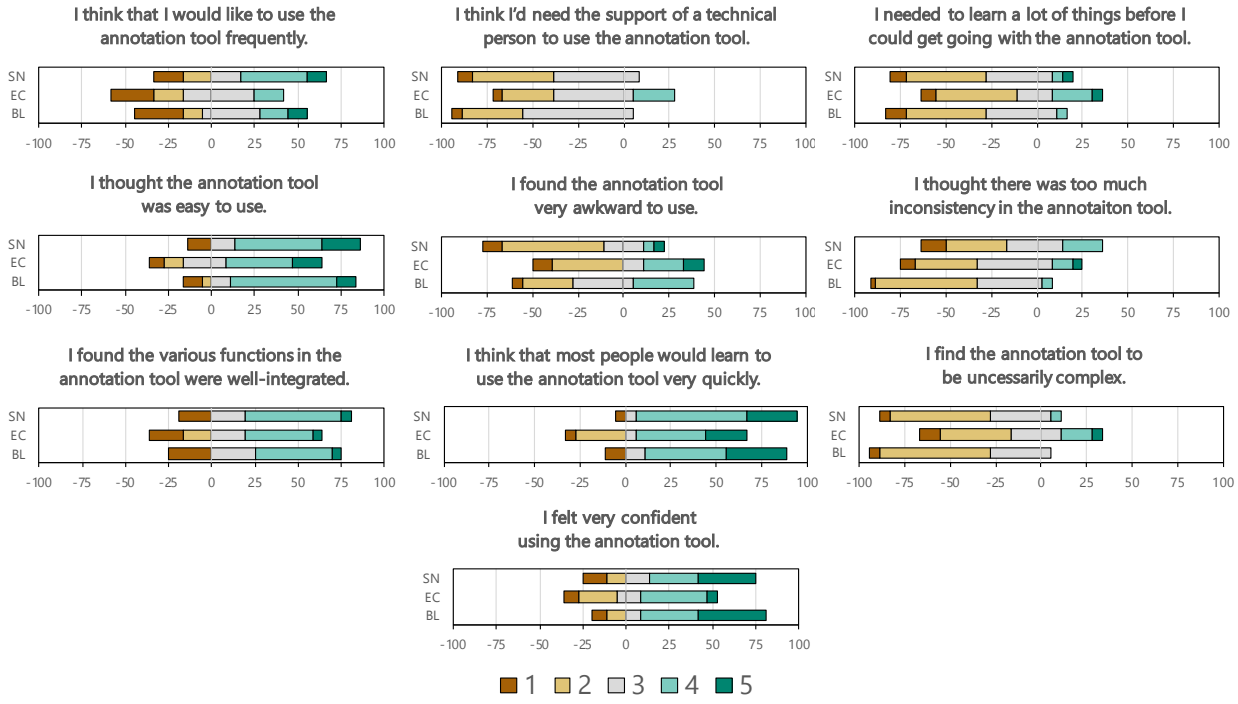


Figure 6: Reported agreement for each of the ten items of the System Usability Scale (SUS) across the three annotation modes. A response of 1 indicates "Strongly Disagree" while a response of 5 indicates "Strongly Agree".

the ability to gain experience with the annotation mode before using it for the task. Following the creation of the labeling jobs, each participant was onboarded with a personalized introduction email that described the nature of the study and a series of study-related steps that detailed the order in which each labeling job or questionnaire should be completed. The email also included an "Annotation Guidelines" PDF document that described the task and provided annotators with three example images that had been annotated with the "correct" annotations. Each training job required annotators to use the job's tool to recreate the annotations shown on the images in the PDF. Participation concluded after the post-study questionnaire was completed.

4.6.3 Study Population. A total of 18 participants were recruited via our snowball sampling approach. Following a balanced study design, a total of three participants were assigned to each configuration shown in Figure 5. All, but one participant reported having at least one year of experience with annotation-related work. 10 participants reported that they they regularly use a computer trackpad as their input device when performing annotation tasks while the remaining 8 reported using a standard computer mouse. 13 participants reported performing bounding box annotation tasks "Sometimes" or "Often". In contrast, 13 participants reported performing annotation tasks with Extreme Clicking either "Never" or "Almost Never".

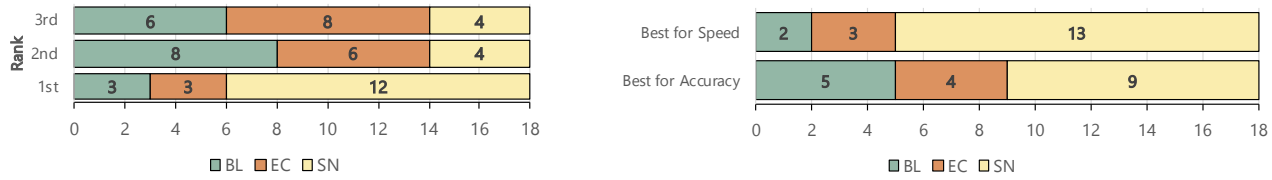
Participants completed the study, on average, in 54 minutes (σ=15.6). Across the five types of questionnaires, the average questionnaire completion time was 46 seconds (σ=15.6). Average responses to Post-Training Questionnaires for the BL (μ=70.7, σ=4.7), EC (μ=70.7, σ=4.7), and SN (μ=70.7, σ=4.7) conditions were significantly positive. The average time spent in the annotation interface was 47 minutes (σ=9.6). Across all three conditions, participants generated 4,900 annotations and 8,086 unique telemetry events related to creating and editing annotations. Similarly, Snapper rendered a total of 8,578 adjustment recommendations during annotation. The average number of recommendations rendered on the annotation canvas in a given drag event (i.e, before the user released their mouse) was 5 (σ=2.7).

5 RESULTS

In this section, we report findings from our "first-use" study with Snapper. To address each of our research questions thoroughly, we structure our findings around the themes that arose from analyzing qualitative data collected across our surveys. Where relevant, we incorporate quantitative data to support observations made via qualitative analysis.

5.1 Supporting Annotation Practices

5.1.1 Understanding First Impressions with Find-and-Snap Tooling. Snapper was generally well-received by participants as a new interactive tool for augmenting how they currently accomplish their annotation work. As P4 notes, "the tool was easy to use, very



(a) Ranking of preferred annotation modes.

(b) Preferred annotation modes for labeling speed and accuracy.

Figure 7: Preferences of annotation modes collected in the Post-Study. "BL" represents Baseline, "EC" represents Extreme Clicking„ and "SN" represents Snapper.

basic, and direct". Snapper received favorable SUS scores (μ=74.8, σ=6.0) which were, on average, slightly higher than both the standard bounding box annotation tool (μ=70.7, σ=4.7) and the Extreme Clicking tool (μ=70.4, σ=4.6). We did not observe a statistically significant difference in scores between annotation modes. However, Friedman tests did reveal statistically significant effects between annotation modes on three items in the SUS: Complexity ($X^2(2) = 6.1$, p=0.046), Inconsistency ($X^2(2) = 0.024$, p=0.002), and Learnability ($X^2(2) = 7.0$, p=0.029). As shown in Table 3, we observe significant differences in Complexity between the Baseline and Extreme Clicking conditions, in Inconsistency between the Snapper and Baseline conditions, and in Learnability between the Snapper and Extreme Clicking conditions. We did not find significant differences for any of the remaining seven dimensions.

Table 3: Results of pairwise Wilcoxon Signed Rank tests on three SUS dimensions.

Dimension	Comparison	*p*	r	
Complexity	Baseline x Extreme Clicking	0.017	0.58	**
Complexity	Snapper x Baseline	1	0.02	
Complexity	Snapper x Extreme Clicking	0.105	0.41	
Inconsistency	Baseline x Extreme Clicking	0.156	0.38	
Inconsistency	Snapper x Baseline	0.004	0.70	**
Inconsistency	Snapper x Extreme Clicking	0.407	0.21	
Learnability	Baseline x Extreme Clicking	0.070	0.45	
Learnability	Snapper x Baseline	1	0.11	
Learnability	Snapper x Extreme Clicking	0.050	0.49	*

We observe similar trends in Complexity, Learnability, and Inconsistency when asking annotators how each annotation mode affected their accuracy or speed. When discussing the Snapper's effect on annotation speed and accuracy, aspects of the tool's ease-of-use were discussed by 10 annotators in light of the tool's ability to "speed up the annotation process" (P15):

> *"It was perfect. I easily understood what to do. I didn't know that it does most of the work for you in terms of refining the box."* - P17 (SN-DM)

Participants provided a number of miscellaneous comments about the system that were by-products of the annotation interface rather than the system itself, such as the interface preventing annotators from dragging an annotation box edge beyond the bounds of an image or rendering adjusted annotations as if they are immutable on rare occasion.

The primary criticism of Snapper's usability was tied to the consistency of its annotation adjustment. 10 participants noted the *frequency* of incorrect adjustments made by Snapper as a factor that influenced their accuracy and speed::

> *"The bounding box made would sometimes shrink incorrectly around a subject. For example, if I put a bounding box around a person, the box would shrink too much and not capture the person's head and arms fully."* - P16 (SN-DM)

Snapper's adjustment behavior was often mentioned by 3 participants when describing positive and negative aspects of other annotation modes. For example, five participants described being "100% in control" (P15, SN-GT) as a key strength for the Baseline annotation mode after having already used Snapper. Even then, sentiments related to Snapper were more positive than the Baseline and Extreme Clicking modes due to Snapper making "the labeling much faster than it would be without it" (P6).

Importantly, annotators experiences with Snapper are grounded in an *existing* annotation mode in which they may have more familiarity and experience. In describing the ease-of-use of the Extreme Clicking mode, 5 participants remarked on the the simplicity of Extreme Point annotation while highlighting a brief learning period:

> *"The [Extreme Point] tool was fairly simple to learn with an easy enough learning curve. I felt I could comfortably use the tool with accuracy within the first few annotation tasks "* - P12 (SN-GT)

Mirroring the demographics of P12, P10 has been engaged in annotation work for more than year, uses a computer mouse for their annotation tasks, and reports "Never" performing Extreme Point annotation tasks. Despite these similarities in demographics, P12 recognizes the value of Extreme Point annotation, but notes a practical challenge in adopting the mode of annotation:

> *"This task was tedious. [Extreme Point] annotation is intuitively simple and feels like it ought to work well. In practice, it might do so, but I think months of practice would be required. I achieved accuracy only through extreme care and slow speed."* - P10 (SN-GT)

Finally, as shown in **Figure 7a**, Snapper's annotation mode was identified by 12 participants as the most preferred mode of annotation in the study. Similarly, as shown in **Figure 7b**, Snapper was reported by the majority of annotators as the annotation mode that enabled them to be both fastest and most accurate with their labeling. In contrast to criticisms related to adjustment inconsitency,

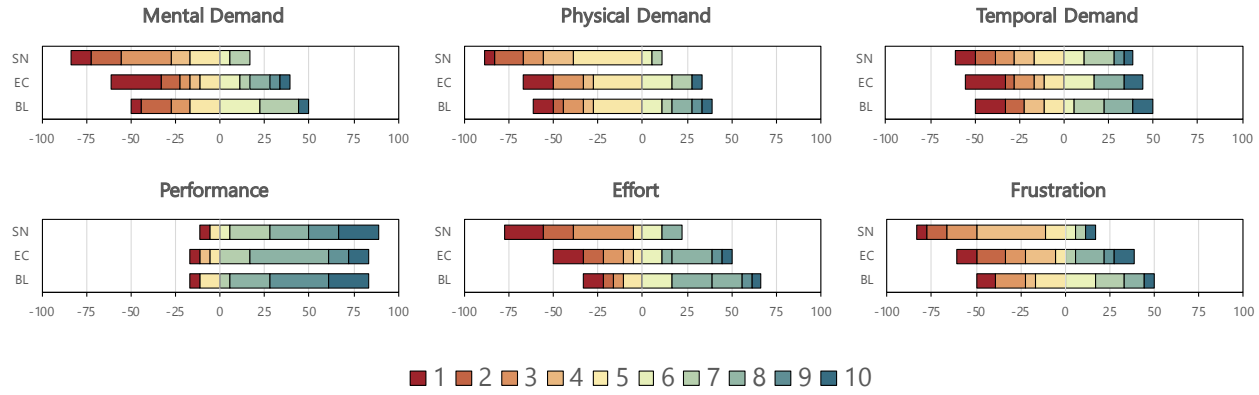


Figure 8: Scores for each the six dimensions of the NASA-TLX across the three annotation modes as observed. A response of 1 indicates "Low" while a response of 10 indicates "High".

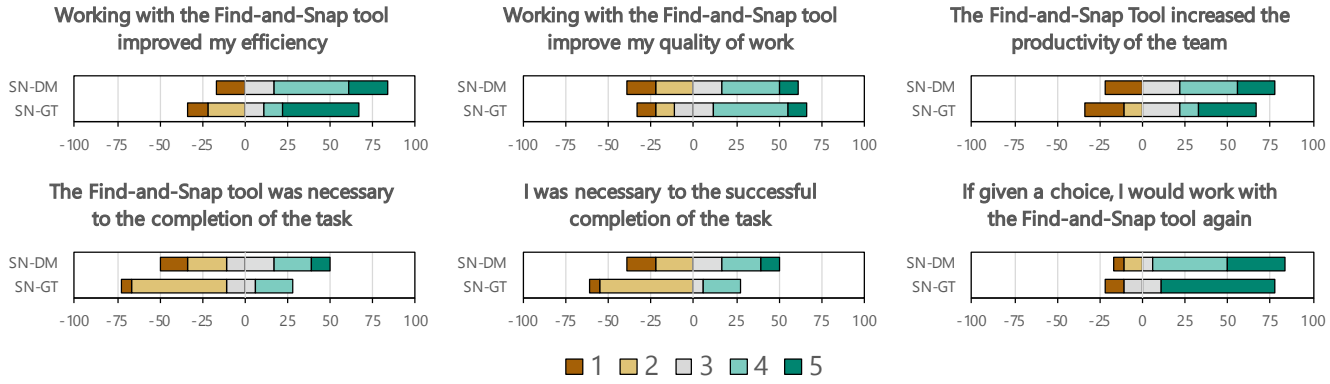


Figure 9: Responses for the Team Productivity and Performance instrument between SN-GT and SN-DM conditions. A response of 1 indicates "Strongly Disagree" while a response of 5 indicates "Strongly Agree".

seven participants noted that the annotations produced by Snapper may require correction, but the correction was usually a smaller amount of work than amount of work required by other annotation modes:

> *"[Snapper was] probably the best of the three. [...] the tool was able to take out some of the preliminary work with the auto-sizing. The auto-sizing didn't always work, but it was easy to correct if it didn't. So, the change was a net positive."* - P13 (SN-GT)

5.1.2 Reducing Cognitive Load in Object Annotation. Snapper's positive first impressions are interlinked with its ability to reduce cognitive effort. For example, Snapper allowed one annotator "to be sloppy/broad about what I was putting in a bounding box"(P16). As shown in **Figure 7a**, the notion of Snapper's ability to reduce effort is supported by a visually observable distinction in NASA-TLX scores between the three annotation modes. Friedman tests revealed statistically significant effects of condition on four of the NASA TLX's six dimensions: Mental Demand ($X^2(2) = 10.3$, p=0.006), Physical Demand ($X^2(2) = 12.9$, p=0.002), Effort ($X^2(2) = 10.1$, p=0.006), and Frustration ($X^2(2) = 8.9$, p=0.01). However, we did not find significant differences for Temporal Demand or Performance. As shown in Table 4, we observe a statistically significant difference exists for each of the four dimensions between Snapper and each of the other two annotation modes. Significant differences in scores between the other two annotation modes were not observed.

5.1.3 Supporting Object Annotation with Interface Intelligence. Snapper introduces interface intelligence into the object annotation context in a way that attempts to offload the necessity of precision. Within this context, several annotators described Snapper's auto-adjustment tool that "covers for me when I slip a bit too far off the mark" (P3). We observe that perceived trust in Snapper (Table 2; Q1) leans positive (μ=3.6; σ=0.9). We specifically find that the average perceived trust is slightly higher for the SN-DM condition (μ=3.8; σ=0.9) in comparison to the SN-GT condition (μ=3.2; σ=0.9), suggesting that Snapper's adjustments made using ground truth data are more mistrusted than those made by Snapper's machine learning model. However, a Mann-Whitney U test did not show a statistically significant difference.

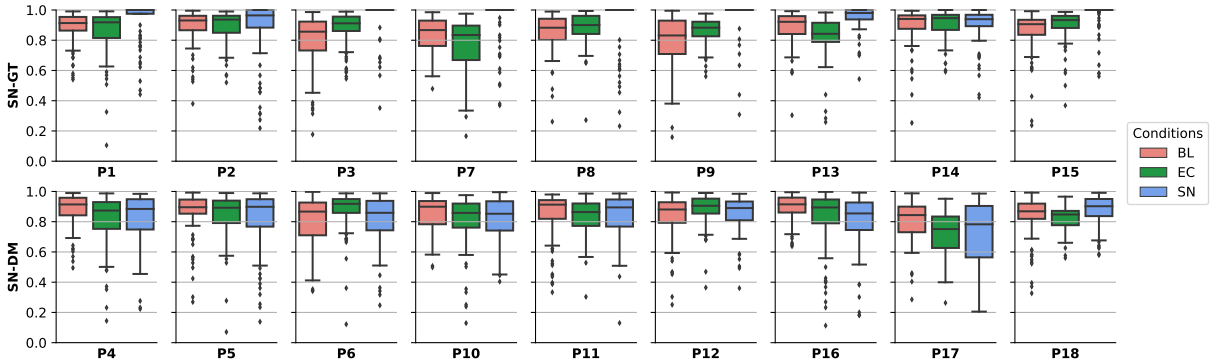


Figure 10: Annotation accuracy (i.e., IoU) across all 18 participants for each condition.

Table 4: Results of pairwise Wilcoxon Signed Rank tests on four NASA-TLX dimensions.

Dimension	Comparison	p	r	
Mental Demand	Baseline x Extreme Clicking	0.535	0.15	
Mental Demand	Snapper x Baseline	0.002	0.70	**
Mental Demand	Snapper x Extreme Clicking	0.018	0.61	**
Physical Demand	Baseline x Extreme Clicking	0.309	0.26	
Physical Demand	Snapper x Baseline	0.001	0.74	***
Physical Demand	Snapper x Extreme Clicking	0.005	0.66	**
Effort	Baseline x Extreme Clicking	0.579	0.1	
Effort	Snapper x Baseline	0.006	0.64	**
Effort	Snapper x Extreme Clicking	0.016	0.57	*
Frustration	Baseline x Extreme Clicking	0.420	0.20	
Frustration	Snapper x Baseline	0.009	0.61	**
Frustration	Snapper x Extreme Clicking	0.008	0.61	**

By analyzing the Team Productivity and Performance measures shown in **Figure 9**, we find that Snapper was generally viewed as a strong complement for task completion. Average agreement responses to statements suggest that participants perceive Snapper as a system that improved personal efficiency (μ=3.8; σ=1.0), improved personal productivity (μ=3.3; σ=1.1), and improved team productivity (μ=3.7; σ=0.7). We also find that participants not only agree that they believe they were necessary for the successful completion of the task (μ=3.8; σ=0.7), but also that they agree that Snapper's as having been necessary for successful completion of the task (μ=2.7; σ=1.0). Further, we observe that participants maintain a desire to work with the tool again (μ=4.1; σ=1.0). Mann-Whitney U tests failed to show a statistically significant difference in responses between SN-GT and SN-DM conditions for all six statements.

5.2 Annotation Quality

To address RQ2, we analyzed annotation quality primarily through a quantitative lens using ground truth labels. Turning first to our participants' qualitative responses about the accuracy of Snapper, we find that the most common response was that Snapper increased the accuracy of their labels. Even for those participants who acknowledged that further adjustments were sometimes required after using Snapper, many felt that these adjustments were often smaller or fewer and farther in between.

> *"I felt like the automatic labeling function was quite accurate. Only a handful of times did I have to adjust the bounding box slightly, and even less times did i have to re-create it around the subject again."* - (P12, SN-DM)

Although noting that accuracy was very good for most objects, a few participants identified cases where Snapper's label accuracy was reduced, including when objects were obstructed or far off in the distance.

> *"Accuracy was excellent for unobstructed people mid-frame. People near the edge of the frame were often reduced to smaller boxes by the tool that couldn't be fixed, hurting accuracy. For whatever reason I found it easier to do manual adjustments with this tool vs the other experimental tools."* - (P10, SN-DM)

With these thoughts in mind, we structure the following section as such. First, we look generally at how Snapper's accuracy compared to the other annotation modes, and then we look at how Snapper performed on objects of varying sizes in comparison to the other annotation modes.

5.2.1 General Annotation Quality. We first turn our attention to **Figure 10** showing annotators' accuracy for all three annotation modes while performing our study. Accuracy was measured as the IoU of participants' annotations with ground truth labels, and annotations with an IoU value of 0 were discarded from this analysis. This was done under the assumption that an IoU value of 0 indicates that the given object was not annotated, and therefore this value does not contribute towards understanding the accuracy of a participants' annotations as it relates to the three modes.

Visually, we see that accuracy is generally high across participants and annotation modes; the average IoU value was 0.84. In terms of difference in annotation quality across conditions, a Kruskal-Wallis test revealed a significant difference in accuracy across the three modes ($X^2(2) = 240$, $p < 0.001$). To find specific differences, we conducted post-hoc Mann-Whitney tests with Bonferroni correction, showing a significant increase in annotation quality for annotations created using Snapper as compared to the baseline bounding box annotation ($p < 0.001$, $r = 0.73$) and Extreme Clicking ($p < 0.001$, $r = 0.76$). However, a Mann-Whitney U test found a significant increase in annotation quality when using the

Table 5: Average IoU scores across the five size labels with standard error reported in parentheses.

	XS	Small	Medium	Large	XL	All
BL	0.66 (0.2)	0.76 (0.2)	0.86 (0.1)	0.86 (0.1)	0.94 (0.0)	0.84 (0.1)
EC	0.65 (0.2)	0.76 (0.2)	0.85 (0.1)	0.85 (0.1)	0.94 (0.1)	0.84 (0.1)
SN-GT	0.87 (0.2)	0.88 (0.2)	0.94 (0.1)	0.97 (0.1)	0.96 (0.1)	0.94 (0.1)
SN-DM	0.58 (0.2)	0.72 (0.2)	0.83 (0.1)	0.90 (0.1)	0.93 (0.1)	0.83 (0.1)
All	0.67 (0.3)	0.77 (0.2)	0.87 (0.1)	0.92 (0.1)	0.94 (0.1)	

SN-GT mode as compared to the SN-DM mode, indicating the possibility that Snapper's improved quality may be largely driven by its Ground Truth iteration.

5.2.2 Annotation Quality by Object Size. Table 5 shows the mean IoU of all participant annotations (except those with an IoU value of 0) binned by object size for each annotation mode. Friedman tests were conducted to determine differences in accuracy between the three larger categories of annotation modes, where SN-GT and SN-DM were taken together to represent the Snapper mode. Altogether, no statistically significant differences were found between the three annotation modes for any of the five defined size groups, suggesting that across the three modes there was generally no difference in annotation accuracy.

Table 6: Mann-Whitney U test results comparing accuracy between SN-GT and SN-DM across object sizes.

Size	U	Z	p	r	
Extra Small	2026	-6.21	< 0.001	0.35	***
Small	15010	-9.04	< 0.001	0.55	***
Medium	74756	-15.21	< 0.001	0.85	***
Large	21864	-11.94	< 0.001	0.66	***
Extra Large	5395	-6.2	< 0.001	0.34	***

Lastly, we compare annotation accuracy between the SN-GT and SN-DM modes. We ran Mann Whitney U tests for comparisons across the five object sizes, and the results are shown in Table 6. The SN-GT mode was found to result in more accurate annotations for every object group as compared to the SN-DM mode.

In sum, although we find no difference in terms of accuracy from the annotation modes across the different object classes represented in this study or different object size groups, we do find that the SN-GT mode consistently outperforms the SN-DM mode for all object sizes. In terms of the IoU values themselves across the object size groups, we see that Snapper in general performs best with larger objects, validating some of our participants' frustrations with using Snapper for smaller objects that are at a distance. This also lends itself towards the recommendation that Snapper be used predominantly for Medium to Large size objects for better annotation quality.

5.3 Accelerating Annotation Time

Despite their familiarity with Bounding Box annotation, many participants consistently reported the baseline tool as decreasing the speed of their annotation, with specific language such as "meticulous" (P14), "a slower workflow" (P2), or involving "a lot of wasted time" (P12) to describe the process. The feelings towards the efficiency of Extreme Clicking were more mixed, with some participants appreciating the simplicity of only having to do 4 clicks to create their annotation, while others found determining where to place these clicks to be more difficult.

> *"I felt this was the fastest of the three tools used. I had no issues being speedy with this tool. The ease of being able to just click to create a point and then have a bounding box be created from those points made this very easy to be quick and accuracte."* - (P2, EC)

> *"I took a lot of time to match each point as correctly as possible. What seems easy to eyeball actually requires making several pixel-length guestimates, then adjusting repeatedly."* - (P10, EC)

On the other hand, our participants largely reported Snapper as helping to improve the speed of their annotation process, combining the familiarity of traditional bounding box annotation with the find-and-snap tooling that allowed for less precision in their initial annotation creation.

> *"This sped up my processing considerably! I was able to annotate images with multiple bounding boxes in seconds, which is super fast."* - (P15, SN-GT)

> *"The tool made the labeling job much much faster. Not needing to be as precise as usual shaved off a few seconds during labeling"* - (P12, SN-DM)

To address RQ3 and explore how well our participants' perceptions of their annotation speed aligned with reality, we analyzed timestamped telemetry data to calculate counts and time spent on annotation activities. We first explore how annotation mode impacted time spent on the annotation task as a whole, and we then specifically focus on annotation mode's impact on time spent creating and editing annotations. A visual representation of annotation times broken down by creation and editing time can be seen in Figure 11.

5.3.1 Analyzing Task Time. For this analysis, we define task time as the temporal difference between the time at which the annotation interface has loaded and the time at which the Submit button was clicked. The average time spent on each task varied across the BL (μ=81.7s; σ=99.3), EC (μ=117.4s; σ=321.4), SN-GT (μ=63.4s; σ=92.2), and SN-DM (μ=47.1s; σ=65.3) conditions. Aggregating SN-GT and SN-DM to represent the Snapper annotation mode, we find that using Snapper led to a 33% reduction in time per task as compared to the BL, while using EC led to an increase of 44% in time per task as compared to the BL.

A Friedman test revealed a statistically significant difference in the time participants spent in completing tasks across the three conditions ($X^2(2) = 99.5$, $p < 0.001$). Post-hoc tests using Wilcoxon tests with Bonferroni correction showed that significant differences in total job time exist between the BL and SN conditions ($p < 0.001$, $r = 0.39$) and between the EC and SN conditions ($p < 0.001$, $r = 0.43$), but not between the BL and EC conditions. Using a Mann-Whitney U test, we also observe a statistically significant difference in task time between the two Snapper conditions ($U = 34{,}630$, $Z = -2.17$, $p = 0.03$, $r = 0.48$). To summarize, among the three annotation modes, participants spent significantly less time on the task when they used Snapper as compared to EC and traditional bounding box annotation. Participants who used SN-DM also spent significantly less time on the task than those who used SN-GT.

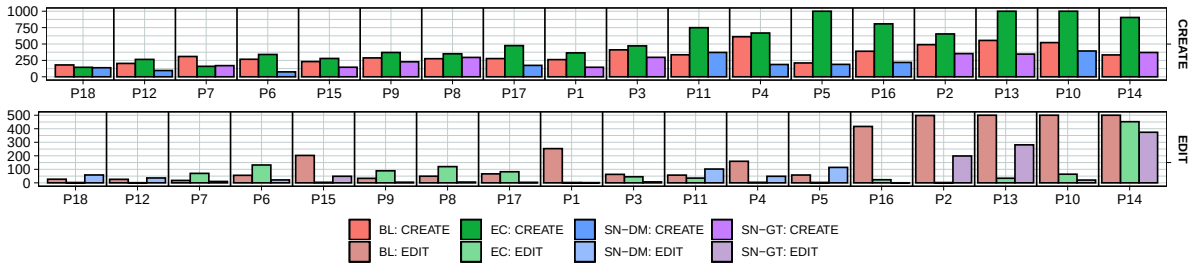


Figure 11: Time spent on annotation creation (`CREATE`) events [Top] and annotation edit (`EDIT`) events [Bottom] across all 18 participants in seconds. Each barplot is shaded per the condition from which it was collected.

5.3.2 Analyzing Annotation Creation Time. We define annotation creation time as the temporal difference between an annotator initiating the creation of an annotation (i.e., mouse-down on the annotation canvas) and completing the creation (i.e., a subsequent mouse-up), and we define cumulative annotation creation time as the sum of all creation time for a given annotation mode. We see differences across the annotation mode between BL (6,155 seconds), EC (11,250 seconds), and Snapper (4,197 seconds). Of this cumulative annotation creation time for Snapper, 56% of it came from participants who used the SN-GT mode. These results show that compared to the BL mode, EC led to an increase in annotation creation time by 87% while Snapper led to a decrease in annotation creation time by 32%.

Average annotation creation time varied across the BL (μ=3.7s; σ=2.5), EC (μ=6.9s; σ=16.8)[2], SN-GT (μ=2.9s; σ=1.9), and SN-DM (μ=2.2s; σ=1.3) conditions. Through a Friedman test, we find that a statistically significant difference exists in the time participants spent creating annotations across the three conditions ($X^2(2)$ = 252.5, p < 0.001). Post-hoc tests using Wilcoxon tests with Bonferroni correction suggest that these differences exist between the BL and EC conditions (p < 0.001, r = 1.0), between the EC and SN conditions (p < 0.001, r = 1.7), and between the BL and SN conditions (p < 0.001, r = 0.9). Using a Mann-Whitney U test, we also observe a statistically significant difference in per-annotation creation time between the two Snapper conditions (U = 430646, Z = -8.63, p < 0.001, r = 0.48). Similar to our results on task time, we find that participants spent less time creating annotations when using Snapper as compared to either the BL or EC, and those who used SN-DM also spent less time creating than those using SN-GT. In addition, participants spent significantly more time creating annotations when using EC as compared to when they used the BL annotation.

5.3.3 Analyzing Annotation Edit Time. We define annotation edit time as the temporal difference between an annotator initiating the modification of an existing annotation (i.e., a mouse-down on the annotation edge) and completing the creation (i.e., a subsequent mouse-up), and we define cumulative annotation edit time as the sum of all edit time for a given annotation mode. We see differences across the annotation mode between BL (3,840 seconds), EC (741 seconds), and Snapper (1,557 seconds). Of this cumulative annotation edit time for Snapper, 75% of it came from participants who used the SN-GT mode. These results show that compared to the BL mode, EC led to a decrease in annotation edit time by 82% while Snapper led to a decrease in annotation edit time by 59%. Additionally, participants who received the SN-GT mode spent a considerably longer amount of time editing their annotations than participants who received the SN-DM mode.

We saw the following average annotation edit times across the BL (μ=2.2s; σ=1.9), EC (μ=2.4s; σ=2.2), SN-GT (μ=2.1s; σ=1.5), and SN-DM (μ=1.5s; σ=0.7) conditions, following a similar trend as annotation creation times. Using similar tests, we observe the presence of a statistically significant difference in the cumulative amount of time spent editing a given annotation across the BL, EC, and SN conditions ($X^2(2)$ = 50.6, p < 0.001). Post-hoc tests using Wilcoxon tests with Bonferroni correction reveal that differences exist between the BL and SN conditions (p = 0.002, r = 0.18), and between EC and SN conditions (p < 0.001, r = 0.21), but not between the BL and EC conditions. As for the differences between the two Snapper conditions, we did not find a statistically significant difference in per-annotation edit time when a Mann-Whitney U test was performed. Similar to per-annotation creation time, we find that per-annotation edit time was reduced when participants used Snapper (and particularly SN-DM) as compared to either the BL or EC annotation modes. This finding is perplexing given that we found that EC had the lowest cumulative time spent for edit events by far, but also has the highest average annotation edit time. However, it can be explained by looking at the total number of events across the three annotation modes. Participants using EC performed far fewer edit actions cumulatively (N = 300) than either BL (N = 1,762) or Snapper (N = 825). Given that less edit events were performed, the average time per edit event for EC is inflated.

6 DISCUSSION

Our study provides insight into the utility of assistive tooling for annotator productivity. Prior work has proposed numerous techniques that are often discussed as being advantageous toward traditional approaches (e.g., standard bounding box annotation) [41, 55, 87]. Here, we find that annotators are receptive to intelligent interface tools that help them to accomplish annotation tasks and increase their productivity. In comparing the system's mode of annotation to alternatives, we find that Snapper enables annotators to accomplish annotation tasks more quickly than other modes of annotation in a way that reduces aspects of cognitive load and does not diminish label quality.

[2]This replicates the finding for average creation time in Papadopoulos et al. [66]

6.1 A Frontier for AI-Assisted Annotation

Our research demonstrates how an established interface technique – snapping – can be interactively intertwined with an object detection model to instill annotator productivity. As a system, Snapper drew inspiration from prior studies that have demonstrated the utility of snapping [10, 11, 49]. Unlike these prior studies, our application of snapping is targeted at the domain of image annotation, which unlike other application domains (e.g., drawing and image editing applications), often comes with constraints (e.g., temporal demands) that can vary from day-to-day for annotators [85]. For these reasons, understanding how Snapper's associated effects and perceived utility shift with longitudinal use is an important area of interest for future work.

Our study provides a narrow example that demonstrates how the design of annotation interfaces can draw from the area of computer-assisted selection. Drawing and image editing applications (e.g., Adobe Photoshop, GIMP, etc.) provide a significant number of referential inspirations for new assistive tool concepts for long, tedious annotation tasks [68]. For example, to what extent might Mortensen and Barrett's *"Intelligent Scissors"* concept [61] be leveraged for assisting annotators in semantic segmentation tasks? How does this concept compare to the state-of-the-art assistive segmentation tools, such as Maninis et al. [55]'s DEXTR? There remains an open frontier for research that explores new tooling for further enhancing annotator productivity with AI-assisted annotation.

6.1.1 Considerations for Annotation Modes. Our study provides a baseline measure of performance for three annotation modes. Contrary to findings from prior research, our findings not only suggest that Snapper's annotation mode yields a smaller amount of annotation time than Papadopoulos et al. [66]'s *Extreme Clicking* approach, but also that traditional bounding box annotation does as well. In 2017, Papadopoulos et al. estimated that annotators require approximately 7 seconds to annotate each object. We replicate their finding by observing that our study's participants yielded an average task time of 6.9 seconds when using Extreme Clicking, which reinforces our quantitative findings regarding how annotators spend their time.

Beyond the notion of tooling, our research carries new design implications for the fundamental nature of annotation modes. Through Snapper, we introduced a new annotation mode named "Find-and-Snap" that automatically adjusted bounding box annotations in real-time. Our demonstration with Snapper is only one example of how such functionality can be facilitated at the level of an "annotation mode". An alternative mode could, for example, leverage Snapper's auto-adjustment feature with Papadopoulos et al. [66]'s Extreme Clicking approach, allowing an adjustment to take place after the forth extreme point has been clicked. For machine learning and interaction design researchers, a wealth of opportunity exists in understanding the strengths and shortcomings of these annotation modes as foundations for machine-assisted support.

6.1.2 Annotation as a Task for Human-AI Collaboration. An important facet of Snapper's interactive design is that it, by design, introduces a new form of *collaboration* between a human annotator and machine learning model. In considering the unique constraints associated with the domain of annotation (i.e., annotators need to move quickly from task to task), we broadly motivated Snapper's design with a select set of established guidelines for engineering Human-AI systems [4, 35]. Despite the system's inability to convey (i.e., graphically, audibly, or in plain text) how or why an annotation will be adjusted, we find that annotators still found significant utility in Snapper. However, we also found that annotators view these capabilities as significantly important. Understanding how to convey how and why an annotation is being adjusted in a manner that is quickly "digestable" to annotators remains an open question.

Despite being beyond the scope of our study, one caveat of introducing Snapper directly into the annotation task is that we may unintentionally introduce a bias into annotators' decision-making processes [31]. Our data suggests that annotators across both Snapper conditions trust the Snapper system, generally believing that the system is capable of producing high-quality adjustments that require little time and effort to further adjust. Despite this, we find that many annotators do, in fact, edit Snapper's adjustments. However, it remains unclear if the annotations created by Snapper's auto-adjustment procedure would mirror the annotations that the annotators would've created without the system's assistance. An important step for subsequent research is examining the development of mental models that annotators develop when using Snapper, giving particular attention to understanding system reliance and the distinctions between Snapper-generated and user-generated annotations. Generally, there is a significant opportunity for exploring approaches that communicate this information in a fashion that is both interpretable and explainable to annotators.

6.2 Considerations for Behavioral Differences

Our study sheds light on the notion that preferences and experience may shape desirability of annotation tools whether they be supported with interface intelligence or otherwise. In exploring how Snapper supported existing image annotation practices, we found a small number of indicators related to individuality that may affect how annotators perceived the utility of the system. Prior work has provided numerous examples for behaviorally profiling annotators in similar ways [16, 33, 73]. Studies of personal productivity highlight the growing importance of supporting peoples' individual practices in their work [39] and engineering systems that ensure their time is well spent [27]. Mark et al. [57] shows that the efficacy of such tools can vary significantly between people.

Through the lens of telemetry, Figure 11 highlights that people may, in fact, differ significantly in the activities (i.e., creation or editing) that they tend to devote their time toward. Snapper's interaction design operates by extending only the annotation creation event, indicating that the system may unintentionally provide support more effectively for annotators who spend most of their time creating annotations. One direction for future work centers on extending Snapper's auto-adjustment feature to edit events (i.e., in which Snapper would recommend adjustments while the corner of an annotation is dragged). Within the broader purview of supporting individual differences, it may be useful to explore how different types of annotators can be automatically inferred (e.g., via telemetry data) such that assistive annotation tools can be used in a more personalized fashion. While additional research is necessary, prior research suggests this is a promising direction [74].

6.3 Limitations

Our study has several limitations. First, our study used images from an established dataset with readily-available ground truth information. We are unable to draw conclusions about the efficacy of Snapper in other dataset contexts that may be distinct in image quality, the number of objects in frame, or domain-specific characteristics. Second, our study was conducted with a unique population of annotators from one company. We are unable to make claims about Snapper's efficacy with other annotator populations that may differ in experience or use alternative input devices (e.g., trackpads). Finally, our findings related to the SN-DM condition are tied Snapper's current object detection model. Though we draw comparisons between Snapper's actual model and a simulated oracle model (i.e., the SN-GT condition), we cannot draw conclusions about the efficacy of using models of greater or lesser performance.

7 CONCLUSION

In this paper, we introduced and evaluated *Snapper*, an interactive and intelligent system that enhances annotator productivity in 2D object detection. tasks with model-generated adjustments. Through a mixed-design user study with full-time annotators, we find that Snapper allows annotators to complete bounding box annotation tasks 39% more quickly than other modes of annotation in a way that requires less cognitive load at no reduction to annotation quality. Further, we observe that annotators perceive Snapper as a tool that is interactively intuitive, trustworthy, and helpful. Through our study, we demonstrate the value in engineering interactive systems that empower annotators in being more productive with automation alongside the challenges that come with it.

REFERENCES

[1] David Acuna, Amlan Kar, and Sanja Fidler. 2019. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11075–11083.

[2] Bishwo Adhikari, Jukka Peltomaki, Jussi Puura, and Heikki Huttunen. 2018. Faster bounding box annotation for object detection in indoor scenes. In *2018 7th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 1–6.

[3] Eirikur Agustsson, Jasper RR Uijlings, and Vittorio Ferrari. 2019. Interactive full image segmentation by considering all regions jointly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11622–11631.

[4] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[5] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[6] Patrick Baudisch, Edward Cutrell, Mary Czerwinski, Daniel C Robbins, Peter Tandler, Benjamin B Bederson, and Alex Zierlinger. 2003. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch-and Pen-Operated Systems.. In *Interact*, Vol. 3. 57–64.

[7] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Adam Eversole. 2005. Snap-and-go: helping users align objects without the modality of traditional snapping. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 301–310.

[8] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. 2016. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*. Springer, 549–565.

[9] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. 2019. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11700–11709.

[10] Eric A Bier. 1990. Snap-dragging in three dimensions. *ACM SIGGRAPH Computer Graphics* 24, 2 (1990), 193–204.

[11] Eric A Bier and Maureen C Stone. 1986. Snap-dragging. *ACM SIGGRAPH Computer Graphics* 20, 4 (1986), 233–240.

[12] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. 2004. Interactive image segmentation using an adaptive GMMRF model. In *European conference on computer vision*. Springer, 428–441.

[13] Yuri Y Boykov and M-P Jolly. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, Vol. 1. IEEE, 105–112.

[14] Virginia Braun and Victoria Clarke. 2012. Thematic analysis. (2012).

[15] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. *European Conference on Computer Vision* (2020).

[16] Ben Carterette and Ian Soboroff. 2010. The effect of assessor error on IR system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 539–546.

[17] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. 2001. A bayesian approach to digital matting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 2. IEEE, II–II.

[18] Cognilytica. 2019. *Data Engineering, Preparation, and Labeling for AI 2020*. Technical Report. https://www.cognilytica.com/document/data-preparation-labeling-for-ai-2020/

[19] Brandon Dang, Miles Hutson, and Matt Lease. 2016. Mmmturkey: A crowdsourcing framework for deploying tasks and recording worker behavior on amazon mechanical turk. *arXiv preprint arXiv:1609.00945* (2016).

[20] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian detection: A benchmark. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 304–311.

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. [n. d.]. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[22] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32, 200 (1937), 675–701.

[23] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM conference on Hypertext and social media*. 218–223.

[24] Michael Gleicher. 1995. Image snapping. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 183–190.

[25] Matthew C Gombolay, Reymundo A Gutierrez, Shanelle G Clarke, Giancarlo F Sturla, and Julie A Shah. 2015. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots* 39, 3 (2015), 293–312.

[26] Anthony G Greenwald. 1976. Within-subjects designs: To use or not to use? *Psychological Bulletin* 83, 2 (1976), 314.

[27] Hayley Guillou, Kevin Chow, Thomas Fritz, and Joanna McGrenere. 2020. Is your time well spent? reflecting on knowledge work more holistically. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–9.

[28] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[29] Björn Hartmann, Leith Abdulla, Manas Mittal, and Scott R Klemmer. 2007. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 145–154.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).

[31] Danula Hettiachchi, Mark Sanderson, Jorge Goncalves, Simo Hosio, Gabriella Kazai, Matthew Lease, Mike Schaekermann, and Emine Yilmaz. 2021. Investigating and Mitigating Biases in Crowdsourced Data. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing*. 331–334.

[32] Danula Hettiachchi, Mike Schaekermann, Tristan J McKinney, and Matthew Lease. 2021. The Challenge of Variable Effort Crowdsourcing and How Visible Gold Can Help. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–26.

[33] Paul Heymann and Hector Garcia-Molina. 2011. Turkalytics: analytics for human computation. In *Proceedings of the 20th international conference on World wide web*. 477–486.

[34] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. 2012. Diagnosing error in object detectors. In *European conference on computer vision*. Springer, 340–353.

[35] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[36] Suyog Dutt Jain and Kristen Grauman. 2013. Predicting sufficient annotation strength for interactive foreground segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1313–1320.

[37] Sanum Joshi. 2019. *How artificial intelligence is creating jobs in India, not just stealing them | India News - Times of India.* . Technical Report. https://timesofindia.indiatimes.com/india/how-artificial-intelligence-is-creating-jobs-in-india-not-just-stealing-them/articleshow/71030863.cms

[38] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. 2018. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision*. Springer, 123–141.

[39] Young-Ho Kim, Eun Kyoung Choe, Bongshin Lee, and Jinwook Seo. 2019. Understanding Personal Productivity: How Knowledge Workers Define, Evaluate, and Reflect on their productivity. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[40] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 1301–1318.

[41] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. 2020. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing* 29 (2020), 7389–7398.

[42] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. 2020. Continuous adaptation for interactive object segmentation by learning from corrections. In *European Conference on Computer Vision*. Springer, 579–596.

[43] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123, 1 (2017), 32–73.

[44] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[45] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. 2020. The open images dataset v4. *International Journal of Computer Vision* 128, 7 (2020), 1956–1981.

[46] Edward Lank and Eric Saund. 2005. Sloppy selection: Providing an accurate interpretation of imprecise selection gestures. *Computers & Graphics* 29, 4 (2005), 490–500.

[47] Gun A Lee and Mark Billinghurst. 2011. A user study on the snap-to-feature interaction method. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 245–246.

[48] Gun A Lee, Ungyeon Yang, Yongwan Kim, Dongsik Jo, and Ki-Hong Kim. 2010. Snap-to-feature interface for annotation in mobile augmented reality. In *Augmented Reality Super Models Workshop at the 9th IEEE International Symposium on Mixed and Augmented Reality*.

[49] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. 2004. Lazy snapping. *ACM Transactions on Graphics (ToG)* 23, 3 (2004), 303–308.

[50] 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 *European conference on computer vision*. Springer, 740–755.

[51] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. 2019. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5257–5266.

[52] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision* (2016).

[53] Yinglu Liu, Hailin Shi, Hao Shen, Yue Si, Xiaobo Wang, and Tao Mei. 2020. A new dataset and boundary-attention semantic segmentation for face parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11637–11644.

[54] Kazuaki Maeda, Haejoong Lee, Julie Medero, and Stephanie Strassel. 2006. A new phase in annotation tool development at the Linguistic Data Consortium: The evolution of the Annotation Graph Toolkit. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*.

[55] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. 2018. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 616–625.

[56] Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* (1947), 50–60.

[57] Gloria Mark, Mary Czerwinski, and Shamsi T Iqbal. 2018. Effects of individual differences in blocking workplace distractions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[58] Anthony May, André Sagodi, Christian Dremel, and Benjamin van Giffen. 2020. Realizing Digital Innovation from Artificial Intelligence.. In *ICIS*.

[59] Mark Maybury. 1998. Intelligent user interfaces: an introduction. In *Proceedings of the 4th international conference on Intelligent user interfaces*. 3–4.

[60] Robert Munro Monarch. 2021. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.

[61] Eric N Mortensen and William A Barrett. 1995. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 191–198.

[62] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 599–606.

[63] Gioacchino Noris, Daniel Sỳkora, Arik Shamir, Stelian Coros, Brian Whited, Maryann Simmons, Alexander Hornung, Marcus Gross, and Robert Sumner. 2012. Smart scribbles for sketch segmentation. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 2516–2527.

[64] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D Wilson. 2016. Snaptoreality: Aligning augmented reality to the real world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1233–1244.

[65] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. 2019. Fast user-guided video object segmentation by interaction-and-propagation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5247–5256.

[66] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. 2017. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*. 4930–4939.

[67] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. 2017. Training object class detectors with click supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6374–6383.

[68] Amy Rechkemmer, Alex C Williams, Matthew Lease, and Li Erran Li. 2023. Characterizing Time Spent in Video Object Tracking Annotation Tasks: A Study of Task Complexity in Vehicle Tracking. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 11. 140–151.

[69] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv* (2018).

[70] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[71] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)* 23, 3 (2004), 309–314.

[72] Joshua S Rubinstein, David E Meyer, and Jeffrey E Evans. 2001. Executive control of cognitive processes in task switching. *Journal of experimental psychology: human perception and performance* 27, 4 (2001), 763.

[73] Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 55–62.

[74] Jeffrey M Rzeszotarski and Aniket Kittur. 2011. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 13–22.

[75] Hisham A Saad, Mark A Terry, Neda Shamie, Edwin S Chen, Daniel F Friend, Jeffrey D Holiman, and Christopher Stoeger. 2008. An easy and inexpensive method for quantitative analysis of endothelial damage by using vital dye staining and Adobe Photoshop software. *Cornea* 27, 7 (2008), 818–824.

[76] Eric Saund and Edward Lank. 2011. Minimizing Modes for Smart Selection in Sketching/Drawing Interfaces. *Sketch-based Interfaces and Modeling* (2011), 55–80.

[77] Robert Simpson, Kevin R Page, and David De Roure. 2014. Zooniverse: observing the world's largest citizen science platform. In *Proceedings of the 23rd international conference on world wide web*. 1049–1054.

[78] Jinming Song and Mohammad Hussaini. 2020. Adopting solutions for annotation and reporting of next generation sequencing in clinical practice. *Practical laboratory medicine* 19 (2020), e00154.

[79] Ivan E Sutherland. 1964. Sketchpad a man-machine graphical communication system. *Simulation* 2, 5 (1964), R–3.

[80] Zsolt Szalavári, Erik Eckstein, and Michael Gervautz. 1998. Collaborative gaming in augmented reality. In *Proceedings of the ACM symposium on Virtual reality software and technology*. 195–204.

[81] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9627–9636.

[82] Carlos Toxtli, Siddharth Suri, and Saiph Savage. 2021. Quantifying the Invisible Labor in Crowd Work. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–26.

[83] Ding Wang, Shantanu Prabhat, and Nithya Sambasivan. 2022. Whose AI Dream? In search of the aspiration in data annotation.. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[84] Zhong-Qiu Wang and Ivan Tashev. 2017. Learning utterance-level representations for speech emotion and age/gender recognition using deep neural networks. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5150–5154.

[85] Alex C Williams, Gloria Mark, Kristy Milland, Edward Lank, and Edith Law. 2019. The Perpetual Work Life of Crowdworkers: How Tooling Practices Increase Fragmentation in Crowdwork. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–28.

[86] Pengfei Xu, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. 2012. Lazy selection: a scribble-based tool for smart shape elements selection. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–9.

[87] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. 2019. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9657–9666.

[88] Weijia Zhang. 2019. *The Status of Chinese Data Annotation Market Needs in 2021 - An industry research report.* Technical Report. https://www.qianzhan.com/analyst/detail/220/210508-8792d1e4.html

[89] Xuefeng Zhang, Bo Liu, Jieqiong Wang, Zhe Zhang, Kaibo Shi, and Shuanglin Wu. 2014. Adobe photoshop quantification (PSQ) rather than point-counting: A rapid and precise method for quantifying rock textural data and porosities. *Computers & Geosciences* 69 (2014), 62–71.