博士論文

Hand-Object Interaction Mining from First-Person Videos

(一人称視点映像からの手-物体インタラクションマイニング)

東京大学大学院
情報理工学系研究科
電子情報学専攻

48-197416　八木 拓真

指導教員　佐藤 洋一　教授

2021 年 12 月

Hand-Object Interaction Mining from First-Person Videos

by

Takuma Yagi

A thesis submitted in partial fulfillment

of the requirements for the degree of Doctor of Philosophy

in Information Science and Technology

Department of Information and Communication Engineering

Graduate School of Information Science and Technology

The University of Tokyo

Committee in charge:

Professor Kiyoharu Aizawa, Chair
Professor Yoichi Sato, Supervisor
Professor Shin'ichi Satoh
Professor Toshihiko Yamasaki
Professor Yusuke Sugano

December 2021

Abstract

Hands are our primary way to interact with the world. Understanding the interaction between human hands and the environment offers valuable insights into fields such as robotics, human-computer interaction, human-robot interaction, and virtual reality. Recently, hand-object interaction understanding from visual inputs has been gaining interest due to the widespread of mobile cameras. Numerous hand-object interaction recognition methods have been developed to recognize the user's short-term actions and spatial configuration of hands and interacting objects. These studies have been conducted in a controlled environment where user action is simple, the target object is evident, and the scene is static. However, the world we live in is far more complicated than we expect. People move around places and perform various actions to meet their needs. Multiple objects are simultaneously involved in an activity and their spatial configuration and appearance change over time by actions performed by the user. While this makes it difficult to even figure out the right object which is in interaction, such real-world aspects have not been taken seriously.

In this thesis, I present methods for recognizing when and for which object the hand-object interaction occurs that generalize to unknown, dynamic, and cluttered scenes in the real world. Specifically, I study the problems of (1) recognizing the contact state between a hand and an object, (2) identifying unique objects appearing in a real-world environment, and (3) their application on assisting users in finding lost objects. Towards developing models that work in real-world environments, they are designed through the unified concept of *hand-object interaction mining*, which comprises the following properties: (i) learning from unlabeled data, (ii) category-agnostic formulation, and (iii) minimum user intervention. Off-the-shelf object detection, tracking, and segmentation techniques are used as a common component for automatically extracting useful knowledge from large-scale unlabeled data. Extensive data collection is conducted for evaluating and discovering unique difficulties that appear in a real-world setting.

In the first work, a method to predict contact states between hands and objects is introduced. Specifically, a video-based method that predicts a sequence of binary contact states (contact or no-contact) from a video and a pair of hand and object tracks is introduced. By predicting hand-object contacts, we can detect objects involved in interactions. However, annotating a large number of hand-object tracks and contact labels is costly. To overcome this difficulty, a semi-supervised framework with two new techniques is introduced: (i) automatic collection of training data with motion-based pseudo-labels and (ii) guided progressive label correction (gPLC) which corrects noisy pseudo-labels with a small amount of trusted data. Because there are no suitable datasets are available for evaluation in real-world environments, a new benchmark on a popular first-person video dataset is introduced. Experiments show that the learned model shows superior performance against existing baseline methods and generalizes well against novel objects and environments.

In the second work, the problem of category-agnostic object instance identification is studied. On understanding hand-object interactions across time, recognizing whether an object is the same one that appeared before will be one of the essential abilities. Because diverse objects appear in real-world environments, it is not realistic to pre-define the target category, and a class-agnostic solution will be demanded. However, no prior works exist on this challenging task, and fundamental difficulties in recognizing object instances in real-world environments were unknown. To this end, a large-scale, challenging benchmark consisting of more than 1,500 unique instances is built on top of unscripted, large-scale first-person videos. Strong metric learning-based baseline models, an in-depth evaluation of the dataset, and a performance comparison against previous datasets are introduced. The analysis shows that the trained model using the created dataset shows better robustness against significant clutters in real-world environments.

In the third work, a practical use-case of hand-object interaction in assisting users in finding lost objects is introduced. People spend an enormous amount of time and effort looking for lost objects. To help remind people of the location of lost objects, various computational systems that provide information on them have been developed. However, prior systems for assisting people in finding objects require users to register the target objects in advance. This requirement imposes a cumbersome burden on the users, and the system cannot help remind them of unexpectedly lost objects. In this study, I propose GO-Finder ("Generic Object Finder"), a registration-free wearable camera-based system for assisting people in finding an arbitrary number of objects based on two key features: automatic discovery of hand-held objects and image-based candidate selection. Given a video taken from a wearable camera, GO-Finder automatically detects and groups hand-held objects to form a visual timeline of the objects. Users can retrieve the last appearance of the object by browsing the timeline through a smartphone app. To investigate how users benefit from using GO-Finder, two user studies are conducted. In the first study, the usefulness of GO-Finder is evaluated by a realistic object retrieval task. In the second study, the system's usability on a longer and realistic scenario is verified, accompanied by an additional feature of context-based candidate filtering. The usefulness of GO-Finder in realistic scenarios where more than one hundred objects appear is verified through experimental results and user feedback.

Acknowledgements

First of all, I would like to express my deepest gratitude to my advisor, Prof. Yoichi Sato, for all the guidance throughout my five-year journey. I was constantly supported by his infinite enthusiasm and perseverance. I learned how to conduct impactful research which appeals to the worldwide community, and how a researcher should behave as a member of society. I would like to also thank Ryo Yonetani, Prof. Yusuke Matsui, and Prof. Ryosuke Furuta for their advice and support from both technical and mental perspectives. Their practical advice made me cultivate my skills in conducting research and writing papers. I also would like to express my gratitude to the thesis committee members: Prof. Kiyoharu Aizawa, Prof. Shin'ichi Satoh, Prof. Toshihiko Yamasaki, and Prof. Yusuke Sugano, for their rigorous review and helpful advices.

None of the achievements could have been achieved without the help of my collaborators. I thank Karttikeya Mangalam for collaborating with me immediately after my enrollment. Thanks to your help, I was able to successfully publish my first research. I thank Chinmoy Samant and Nikita Kister for spending time with me for research discussion and collaboration. I would like to thank Kunimasa Kawasaki for the long-standing collaboration over years. I have learned broad knowledge on developing a practical system with edge devices, and the discussion made with him helped me nourish the big picture of my research. I thank Moe Matsuki for giving me fresh ideas when I feel stuck and less confident.

I also would like to thank all the lab members who collaborated with me during the five years in Sato Lab. I thank Keita Higuchi for the fruitful moments in casual discussion, leading me to enter the field of Human-Computer Interaction. I thank Rie Kamikubo for collaborating with me on visual navi-

gation. It was a pleasure to learn how difficult to apply fundamental research to the field. I thank Takumi Nishiyasu for constantly helping me in various situations in both research and lab management. I thank Md. Tasnimul Hasan willingly spared his time for an extensive amount of annotation. I thank Takehiko Ohkawa for giving me the passion to proceed forward and for a fruitful research discussion.

I also thank all the members of WBA Future Leaders for helping me foster a wide perspective on how to tackle the broad field of Artificial Intelligence. I especially would like to thank Masahiko Osawa for expecting me and allowing me to be a leader of this group. While I can't list up all of the members' names, I thank all of you for sharing the time with me during the events.

I would also like to tell my gratitude to Masason Foundation and The University of Tokyo Toyota-Dwango Scholarship for Advanced AI Talents for their financial support for living and research.

Finally, I express my immeasurable gratitude to my family for their unconditional love and everlasting support.

December 2021

Takuma Yagi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Hands are one of our primary ways to interact with the world. Understanding the interaction between human hands and environment offers several important applications such as action prediction [2, 3], rehabilitation measurements [4], knowledge transfer for robot manipulation [5], and virtual reality [6]. Hand-object interaction understanding from visual sensors has been gained interest because it does not require the person to wear an external device to their hand nor attach sensors to the object of interest. To capture hand-object interaction from visual inputs, first-person (egocentric) perception [7], which captures video footage from a body-worn camera is found to be promising since it captures the interacting hand and object up-close compared to the traditional third-person cameras. Because the wearable camera moves along with the user, it can efficiently capture the user's action and its surrounding environment instead of placing multiple sensors in the environment. In addition to its advantage on sensor placement, this human-centric perspective is suitable for tracking continuous, everyday interaction for a long-term duration. By integrating wearable cameras with an intelligent recognition system, we can realize a future of people receiving useful feedback from the system by monitoring the current status of the wearer and environment [8–10].

Recently, numerous hand-object interaction recognition methods have been developed to recognize the user's short-term actions and spatial configuration of hands and interacting objects [11]. While there has been significant progress in tasks such as 3D hand pose estimation [12], and reconstruction of the 3D hand and its interacting objects [13–15], they have been typically studied in controlled settings where user actions are simple, the object of interest is evident, and the scene remains static [13, 16–19]. Limited numbers of objects and actions are carefully selected for the study, and the evaluations are conducted in similar environments as training.

However, the real world we live in is far more complicated than we expect. Users move around places and perform various actions to meet their needs. Their actions are unscripted, rapid, and often subtle. Many objects are simultaneously involved in an activity, and their spatial configuration and appearance change over time by the user's action. While the above real-world aspects make it difficult to even figure out the right object which is in interaction, such aspects had not been taken seriously. In the above context, a method generalizes to unknown and cluttered scenes in the real world is demanded.

In this thesis, I present fundamental methods for recognizing *when* and for *which object* the hand-object interaction occurs that generalize to unknown, dynamic, and cluttered scenes in the real world. Specifically, I study the problems of (1) recognizing the contact state between a hand and an object, (2) identifying unique objects appearing in a real-world environment, and (3) their application on assisting users in finding lost objects. Towards developing models that work in real-world environments, I propose a unified concept named *hand-object interaction mining*, consisting of three principles:

Learning from unlabeled data Hand-object interaction understanding typically requires an extensive amount of annotation (*e.g.*, the spatial configuration of hand and object, their interaction state, and their changes over time). However, it is unrealistic to provide full supervision to a large number of examples because of the high annotation cost. Therefore, we utilize the off-the-shelf object detection and tracking method to extract valuable information from unlabeled data without manual supervision.

Category-agnostic formulation Most visual recognition models exploit the "closed world assumption", *i.e.*, the presumption that a true statement is also known to be true. For example, in an object recognition system, the object categories to be recognized are limited to pre-defined categories and images other than that categories will be ignored as irrelevant or false input. However, arbitrary types of objects will appear in the real world. As long as it is a hand-manipulated object, it must be included in the processing target. In addition, the objects' form and appearance dynamically change across time through interactions. For example, in the cooking domain, the form of the food significantly changes by the cooking action (*e.g.*, chop, stir, and heat). The appearance of a mug significantly changes by which drink to be poured to. Under such atypical appearance change, it will be difficult to cover enough data variation for all the object categories. Instead of designing a model specialized for a limited number of categories, a model should be designed to be category-agnostic—arbitrary objects should be treated equally without filtering them.

Minimum user intervention On deploying a system to users, it is important not to give unnecessary burden to the user to use it. For example, excessive inquiries to the user will significantly impair the user experience. Therefore, the system should automatically collect information around the user as possible instead of asking unnecessary inquiries.

All the methods in this thesis are developed by following the above three principles. Instead of manually collecting data, off-the-shelf object detection, tracking, and segmentation techniques trained by recent large-scale video datasets [20–23] are effectively used as a common component for efficiently extracting useful knowledge from large-scale unlabeled data. In addition, extensive data collection is made to evaluate the proposed methods in real-world environments.

1.2 Overview and Organization

The overview of this thesis is given as follows:

1. Hand-Object Contact Prediction (Chapter 2)

First, I work on the task of predicting the contact state between hands and objects. In determining which hand-objects pair are interacting, it is important to infer when hands and objects are in contact. Despite its importance, prior works on action recognition typically modeled action at a video clip level and were not modeled as an interaction between a specific pair of hands and objects. Towards more precise modeling of hand-object interaction, a video-based method for predicting contact between a hand and an object is introduced. Specifically, given a video and a pair of hand and object tracks, the model predict a binary contact state (contact or no-contact) for each frame. However, annotating a large number of hand-object tracks and contact labels against the diverse environment is costly. To overcome this difficulty, a semi-supervised framework with two new techniques is introduced: (i) automatic collection of training data with motion-based pseudo-labels and (ii) guided progressive label correction (gPLC) which corrects noisy pseudo-labels with a small amount of trusted data. Because there were not suitable datasets available for contact prediction in real-world environments, a new benchmark is built for evaluation. Experiments show that the learned model shows superior performance against existing baseline methods and generalizes well in the case of novel objects and environments.

2. Category-Agnostic Object Instance Identification (Chapter 3)

In this chapter, a further look at the object side of hand-object recognition will be conducted. On understanding long-term hand-object interactions across time, recognizing whether an object is the same one that appeared before will be one of the essential abilities. This object instance identification ability contributes to multiple applications such as object state-change recognition, long-term action understanding, and user assistance by tracking the presence of objects. In the real world, arbitrary types of object instances will appear and it is not ideal to limit the object categories to be recognized in advance. Therefore, a class-agnostic solution is demanded considering practical use. In addition, each instance should be discovered without presuming its existence in a novel environment.

To this end, I work on the challenging problem of category-agnostic object instance identification. It is formulated as clustering of object image tracks that appear in videos. Because no suitable dataset exists for evaluation, *EK-Instance*, a large-scale, challenging benchmark consisting of more than 1,500 unique instances is built upon the EPIC-KITCHENS dataset [23]. To extract fundamental difficulties in the dataset, strong metric learning-based models, an in-depth evaluation of the dataset, and a performance comparison against previous datasets are introduced. The analysis shows that the trained model using the created dataset shows better robustness against significant clutters in real-world environments.

3. Assisting Users in Finding Lost Objects (Chapter 4)

In this chapter, I present a practical use-case of hand-object interaction in assisting users in finding lost objects. People spend an enormous amount of time and effort looking for lost objects. To help remind people of the location of lost objects, various computational systems that provide information on their locations have been developed. However, prior systems for assisting people in finding objects require users to register the target objects in advance. This requirement imposes a cumbersome burden on the users, and the system cannot help remind them of unexpectedly lost objects. In this study, I propose GO-Finder ("Generic Object Finder"), a registration-free wearable camera-based system for assisting people in finding an arbitrary number of objects based on two key features: automatic discovery of hand-held objects and image-based candidate selection. Given a video taken from a wearable camera, GO-Finder automatically detects and groups hand-held objects to form a visual timeline of the objects. Users can retrieve the last appearance of the object by browsing the timeline through a smartphone app. I conducted user studies to investigate how users benefit from using GO-Finder. In the first study, I asked participants to perform an object retrieval task and confirmed improved accuracy and reduced mental load in the object search task by providing clear visual cues on object locations. In the second study, the system's usability on a longer and realistic scenario was verified, accompanied by an additional feature of context-based candidate filtering.

Participant feedback suggested the usefulness of GO-Finder also in realistic scenarios where more than one hundred objects appear.

Chapter 2

Hand-Object Contact Prediction

2.1 Introduction

Recognizing how hands interact with objects is crucial to understand how we interact with the world. Hand-object interaction analysis contributes to several fields such as action prediction [2], rehabilitation [4], robotics [24], and virtual reality [6].

Every hand-object interaction begins with contact. In determining which hand-object pairs are interacting, it is important to infer when hands and objects are in contact. However, despite its importance, finding the beginning and the end of hand-object interaction has not received much attention. For instance, prior works on action recognition (*e.g.*, [25]) attempt to recognize different types of hand object interactions at the video clip level, i.e., recognizing one action for each video clip given as input. Some other works on action localization (*e.g.*, [26]) can be used for detecting hand object interactions but localized action segments are not necessarily related to the beginning and the end of contact between hands and objects. Contact between a hand and an object has been studied in the context of 3D reconstruction of hand object interaction [13, 14]. However, they assumed that hands and objects are already interacting with each other. Only the moment when hands and objects are interacting in a stable grasp was targeted for

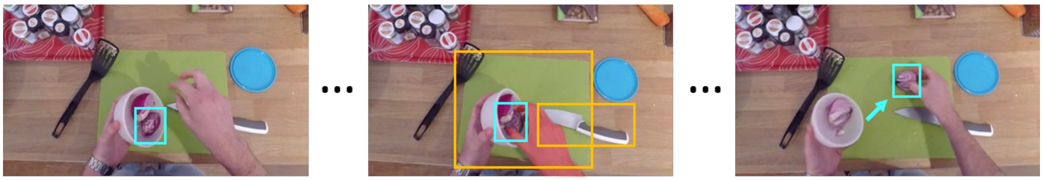


Figure 2.1: **Overlap on Image = Contact?:** Right hand (masked red region) grabs the onion (middle). Surrounding objects (cutting board, knife) overlaps with hand but not in contact. While it is difficult to determine contact state from single image, we can ease the problem by looking at temporal context (left and right).

analysis.

In this work, the task of predicting contact between a hand and an object from visual input is studied. Predicting contact between a hand and an object from visual input is not trivial. For example, even if the hand area and the bounding box of an object overlap, it does not necessarily mean that the hand and the object are in contact (see Figure 2.1). In determining whether a hand and an object are in contact, it is essential to consider the spatiotemporal relationship between them. While some methods claim that the hand contact state can be classified by looking at hand shape [22, 27], they did not explicitly predict the contact state between a specific pair of a hand and an object, limiting their utility. This work aims to fill the gap between utilization of hand presence and detailed 3D understanding by predicting whether a hand is in contact with an object.

Specifically, a video-based method for predicting binary contact states (contact or no-contact) between a hand and an object in every frame is proposed. We assume tracks of hands and objects specified by bounding boxes (*hand-object tracks*) as input, and infer the contact state between the specified hand-object pair. However, annotating a large number of hand-object tracks and their contact states can become too costly. To overcome this difficulty, a semi-supervised framework consisting of (i) automatic training data collection with motion-based pseudo-labels and (ii) guided progressive label correction (gPLC) which corrects noisy pseudo-labels with a small amount of trusted data is introduced.

Given unlabeled videos, off-the-shelf detection and tracking models are applied to form a set of hand-object tracks. Then pseudo-contact state labels to each track are assigned by looking at its motion pattern. Specifically, we assign a contact label when a hand and an object are moving in the same direction and a no-contact label when a hand is moving alone.

While generated pseudo-labels can provide valuable information on determining the state of contact states with various types of objects when training a prediction model, the pseudo-labels also contain errors that hurt the model's performance. To alleviate this problem, those errors are corrected by the guidance of an additional model trained on a small amount of trusted data. In gPLC, two networks each trained with noisy labels and trusted labels are trained. During the training, noisy pseudo-labels are iteratively corrected based on both network's confidence scores. A small amount of trusted data is used to guide which label to be corrected and yield reliable training labels for automatically extracted hand-object tracks. A novel contact state prediction model which combines appearance and motion information, which will be trained with gPLC is also presented.

Because there was no benchmark suitable for this task, I newly annotated contact states to various types of interactions appearing in the EPIC-KITCHENS dataset [23, 28] which includes in-the-wild cooking activities. I show that the prediction model achieves superior performance against frame-based models [22, 27], and the performance further boosted by using motion-based pseudo-labels along with the proposed gPLC scheme.

The contributions include: (1) A video-based method of predicting contact between a hand and an object leveraging temporal context; (2) A semi-supervised framework of automatic pseudo-contact state label collection and guided label correction to complement lack of annotations; (3) Evaluation on newly collected annotation over a real-world dataset.

2.2 Related Work

Reconstructing hand-object interaction Reconstruction of the spatial configuration of hands and their interacting objects plays a crucial role to understand hand-object interaction. 2D segmentation [29] and 3D pose/mesh estimation [13, 19, 30–32] of hand-object interaction were studied actively in recent years. They aims to estimate either hand and object pose directly or otherwise estimate the parameters of the hand model. In addition to 3D shape, richer information could be obtained from synthetic data [13, 33], whole-body hand-object interaction capture [34], thermal sensor [17, 18], and 3D object models [14]. Brahmbhatt *et al.* [17] propose ContactDB, which obtains ground-truth contact region of an object using a thermal sensor. Recently, Cao *et al.* [14] achieved in-the-wild 3D hand-object reconstruction by introducing depth and penetration constraint. However, they assume (1) 3D CAD models exist for initialization (except [13]) (2) the hand is interacting with objects, making the methods inapplicable when hand and object are not interacting with each other. While multiple datasets appear for hand-object interaction analysis [16, 17, 22, 29], no dataset focused on the entire process of interaction including beginning and termination of contact. It is worth mentioning DexYCB [35], which captured sequences of picking up an object. However, the performed action was very simple and their analysis focused on 3D pose estimation rather than contact modeling between hands and objects. They assumed interaction mining is done and only captured actions where hands and objects are already in contact. We study the front stage of the hand-object reconstruction problem—whether the hand interacts with the object or not. To avoid erroneous mesh prediction and optimization, we resort to 2D-level inference using appearance and motion cues.

Hand-object contact prediction Contact prediction is found to be a difficult problem because contact cannot be directly observed due to occlusions. To avoid using intrusive hand-mounted sensors, contact and force prediction from visual input was studied [34, 36–38]. For example, Pham *et al.* [36] present an RNN-based force estimation method trained on force and kinematics measurements from force transducers. Ehsani *et al.* [38] obtained supervision from a simulator to infer contact and force from video. Taheri *et al.* [34] collected whole-body grasps with contact annotation using high-precision motion capture, enabling to generate realistic grasps against novel object. Other than hand-object interaction, whole-body force [39], ground contact [40], and foot pressure [41] prediction is studied. These methods require a careful setup of sensors, making it hard to apply them in an unconstrained environment.

Instead of precise force measurement, a few methods study contact state classification (*e.g.*, no contact, self contact, other people contact, object contact) from an image [22, 27]. Shan *et al.* [22] collected a large-scale dataset of hand-object interaction along with annotated bounding boxes of hands and objects in contact. They train a network which detects hands and their contact state (no contact, self contact, other person contact, portable object contact, and static object contact) from its appearance. Narasimhaswamy *et al.* [27] extends the task into multi-class prediction. While their formulation is simple, they did not take the relationship between hands and objects explicitly and were prone to false-positive prediction. To balance utility and convenience, we take the middle way between the two approaches—binary contact state prediction between a hand and an object specified by bounding boxes.

Learning from noisy labels Since dense labels are often costly to collect, methods to learn from large unlabeled data are studied. While learning features from weak cues are studied in object recognition [42] and instance segmentation [43], it was not well studied in a sequence prediction task. While automatically-generated labels mitigates small number of labeled examples, generated pseudo-labels typically include noise which harms the model's performance. Various approaches such as loss correction [44, 45], label correction [46, 47], sample selection [48], and co-teaching [49, 50] are proposed to deal with noisy labels. Loss correction and sample selection aims to modify the loss function to eliminate the effect of noisy labels by either estimating the noise transition matrix or weighting across samples. On the other hand, label correction explicitly identify and fixes the noisy label based on own network's prediction. Co-teaching propose to train two networks supervised by each other to avoid overfitting to noisy labels. However, most methods assume feature-independent feature noise which is over-simplified, and only a few works study realistic feature-dependent label noise [47, 51]. Zhang *et al.* [47] propose progressive label correction (PLC) which iteratively corrects labels based on the network's confidence score with theoretical guarantees against feature-dependent noise patterns. Inspired by PLC [47], we propose gPLC which iteratively corrects noisy labels by not only the prediction model but also with the clean model trained on small-scale trusted labels.

2.3 Proposed Method

In contrast to prior works [22, 27] we formalize the hand-object contact prediction problem as predicting the contact states between a hand and a specific object appearing in a image sequence. We assume video frames $\mathcal{X} = \{X^1, \ldots, X^T\}$, hand instance masks $\mathcal{H} = \{H^1, \ldots, H^T\}$, and target object bounding boxes $\mathcal{O} = \{O^1, \ldots, O^T\}$ as inputs, forming a hand-object track $\mathcal{T} = (\mathcal{X}, \mathcal{H}, \mathcal{O})$.

Our goal is to predict a sequence of a binary contact state ("no contact" or "contact") $\mathbf{y} = \{y^1, \ldots, y^T\}(y \in \{0, 1\})$ given a hand-object track $\mathcal{T}$. If any physical contact between the hand and the object exists, the binary contact state y is set to 1, otherwise 0. Although we do not explicitly model two-hands manipulation, we consider the presence of another hand as side information (see Section 2.3.3 for details).

However, collecting a large number of hand-object tracks and contact states for training can become too costly. We deal with this problem by automatic pseudo-label collection based on motion analysis and a semi-supervised label correction scheme.

2.3.1 Pseudo-Label Generation from Motion Cues

We automatically detect hand-object tracks and assign pseudo-labels to them based on two critical assumptions. (i) When a hand and an object are in contact, they exhibit similar motion pattern. (ii) When a hand and an object are not in contact, the hand moves while the object remains static (see Figure 2.2 left for illustration). Because these assumptions are simple yet applicable regardless of object appearance and motion direction, we can use these motion-based pseudo-labels for training to achieve generalization against novel objects.

Hand-object track generation Given a video clip, we first use the hand-object detection model [22] to detect bounding boxes of hands and candidate objects appearing in each frame. Note that the detected object's contact state is unknown and objects which overlap with hands are detected. For each hand detection, we further apply a hand segmentation model trained on EGTEA dataset [20] to each hand detection to obtain segmentation masks.

Next, we associate adjacent detections using a Kalman Filter-based tracker [52]. However, since [22] does not detect objects away from the hand, we extrapolate object tracks one second before and after using a visual tracker [53], producing $\mathcal{H}$ and $\mathcal{O}$. Finally, we construct the hand-object track $\mathcal{T}$ by looking for pairs of hand and object tracks which include a spatial overlap between hand mask and object bounding box.

Contact state assignment We find contact (and no-contact) moments by looking at the correlation between hand and object motion. First, we estimate optical flows between adjacent frames. Since we are interested in relative movement of hands and objects against backgrounds, we obtain background motion-compensated optical flow and its magnitude M by homography estimation. Specifically, we sample flow vectors outside detected bounding boxes as matches between frames and estimate the homography using RANSAC [54].

Let $F = (f_{ij}) = \mathbb{I}_{(M>\sigma)}$ be a binary mask of foreground moving region its magnitude larger than a certain threshold σ. For each hand and object binary region mask $H = (h_{ij})$ and $O = (o_{ij})$, we calculate the ratio of moving region within each region: $h_r = \frac{\sum_{ij}(h_{ij} \cdot f_{ij})}{\sum_{ij} h_{ij}}$, $o_r = \frac{\sum_{ij}(o_{ij} \cdot f_{ij})}{\sum_{ij} o_{ij}}$. We assign a label to a frame if $\mathrm{IoU}(H, O) > 0$ and h_r and o_r above certain thresholds. Similarly, we assign a no-contact label if $\mathrm{IoU}(H, O) = 0$ or h_r above threshold but o_r below threshold. However, the above procedure may wrongly assign contact labels if the motion direction of hand and object are different (*e.g.*, the object handled by the other hand). Thus we calculate the cosine similarity between the average motion vector of hand and object region and assign a contact label if above threshold otherwise a no-contact label. To deal with errors in flow estimation, we cancel the assignment if the background motion ratio $b_r = \frac{\sum_{ij}(b_{ij} \cdot f_{ij})}{\sum_{ij} b_{ij}}$ ($B = (b_{ij})$ denotes background mask other than H and O) is

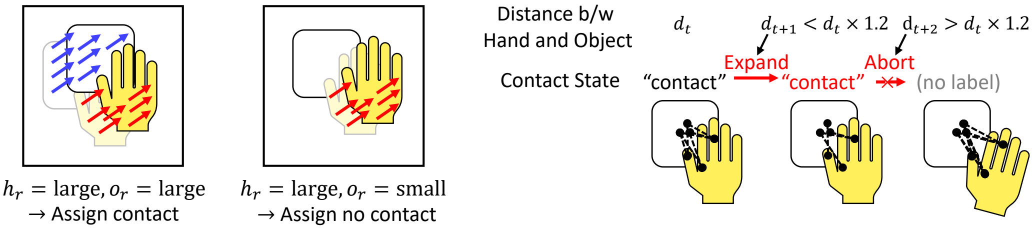


Figure 2.2: (Left) Pseudo-label generation from motion cues. (Right) Pseudo-label extension based on hand-object distance.

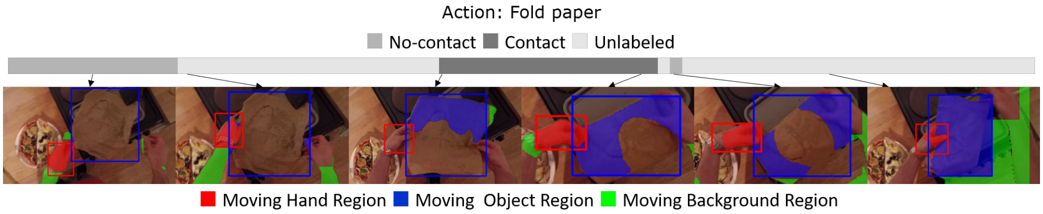


Figure 2.3: **Example of generated pseudo-labels:** (Top) Gray and dark gray bar indicates no-contact/contact labels otherwise no labels assigned. (Bottom) Representative frames. Red, blue, and green regions denote moving hand, object, and background regions, respectively. In rightmost frame, no label is assigned because of abrupt background motion.

above threshold.

Pseudo-label extension Based on the above procedure, we obtain pseudo-labels partially assigned on hand-moving frames. The above procedure assigns labels on hand-moving frames, but it does not assign labels when hands are moving slowly or still. To assign labels also on those frames, we extend the assigned contact states if the relationship between hands and objects does not change from the timing when pseudo-labels are assigned (see Figure 2.2 right).

To track hand-object distance, we find point trajectories from hand and object region which satisfy forward-backward consistency [55]. We then calculate the distance d between each hand-object point pair and compare the average distance of them in each frame. We extend the last contact state if the average distance is within a certain range of that of the starting frame.

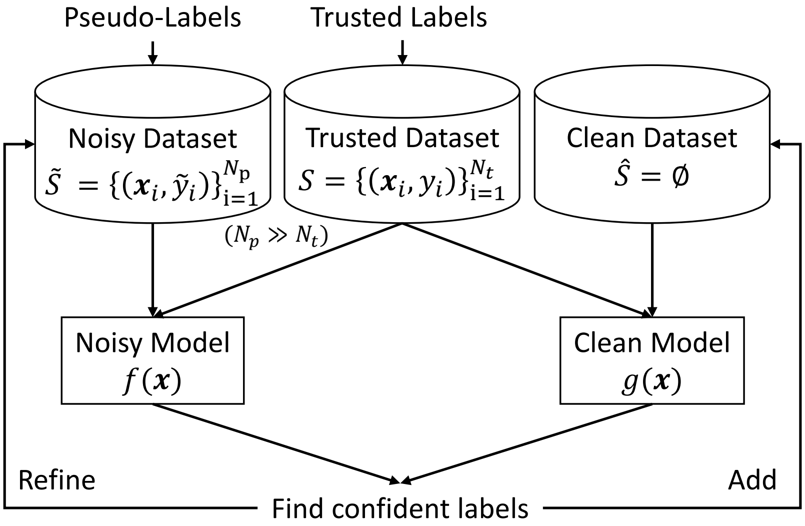


Figure 2.4: Overview of guided progressive label correction (gPLC).

Figure 2.3 shows an example of the generated pseudo-labels.

2.3.2 Guided Progressive Label Correction (gPLC)

While generated pseudo-labels include useful information in determining contact states, they also include errors induced by irregular motion patterns. The model may overfit to noise if we simply train it based on these noisy labels. To utilize reliable labels from noisy pseudo-labels, we propose a semi-supervised procedure called guided progressive label correction (gPLC), which works with a small number of trusted labels. We summarize the procedure in Algorithm 1. We assume a small number of trusted labels for the rescue to guide which label to correct the pseudo-labels.

We assume a noisy dataset $\tilde{S}$ with generated pseudo-labels and a trusted dataset S with manually annotated trusted labels. We train two identical networks, each called noisy model and clean model. The noisy model f is trained on both $\tilde{S}$ and S while the clean model g is trained on S and a clean dataset $\hat{S}$ which is introduced later. We perform label correction against generated pseudo-labels in $\tilde{S}$ using the prediction of both models.

As training of f proceeds, it will generates input region with high confidence against them. Similar to PLC [47], we try to correct labels on which

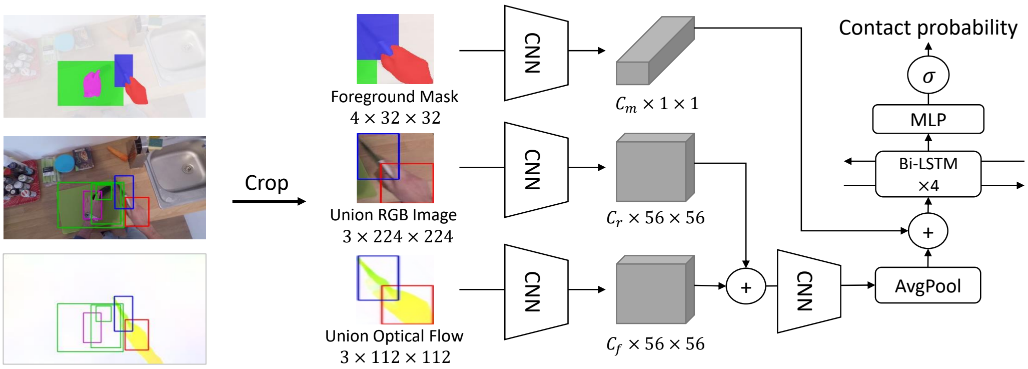


Figure 2.5: Architecture of contact prediction model.

f gives high confidence. Note that we correct labels in a frame-wise manner, assuming output contact probability is produced per frame. In gPLC, we correct labels only when f has high confidence and does not contradict the clean network g's prediction. Because $\tilde{S}$ is generated from motion cues, the decision boundary of f may be different from that of the optimal classifier. Thus the label correction on f alone would not converge to the desired decision boundary. Therefore, we guide the correction process by using g, which is trained on small-scale but trusted data. Starting with a strict threshold on δ, we iteratively correct labels upon training. When the number of corrected labels gets small enough, we increase δ to loosen the threshold and continue the same procedure. However, since g is trained on a small-scale data, it has the risk of overfitting to S. To prevent overfitting, we iteratively add data that f gives high confidence to another dataset called clean dataset $\hat{S}$ and feed them to g so that g also grows through training. Initially $\hat{S}$ will not contain labels, but high-confident labels will be added over time. See Algorithm 1 for detail. In implementation, $f(\mathbf{x})$ and $g(\mathbf{x})$ are trained beforehand by $\tilde{S}$ and S before starting the gPLC iterations.

2.3.3 Contact Prediction Model

To capture the spatial relationship of hands and objects, we propose an RNN-based method that takes RGB images, optical flow, and mask information as input (see Figure 2.5). An overview of the model for a single frame of input is shown in Figure 2.5. For each modality, we crop the input by taking the union of the hand region and the object bounding box. The optical flow is a three-channel image consisted of x-axis motion, y-axis motion, and magnitude. The foreground mask is a four-channel binary mask that tells the presence of a target hand instance mask, a target object bounding box, other detected hand instance masks, and other detected object bounding boxes. The former two channels specify which region to attend, the latter two channels prevent confusion when the target hand or object interact with other entities. RGB and flow images are fed into each encoder branch, concatenated at the middle, and then passed to another encoder. Both encoders consist of several convolutional blocks, each consisted of 3×3 convolution followed by a ReLU and a LayerNorm layer [56], and a 2×2 max-pooling layer to reduce the spatial resolution. The foreground mask encoder consists of three convolutional layers each followed by a ReLU layer, producing a 1×1 feature map encoding the positional relationship between the target hand, the target object, and the other hands and objects. After concatenating the features extracted from the foreground mask, contact probability is calculated through four bi-directional LSTM layers and three layers of MLP.

Training Objective We train the network by a standard binary cross-entropy loss weighted by the ratio of the amount of labels in the training data. We did not propagate the error for non-labeled frames.

2.4 Experiments

Since there was no benchmark suitable for our task, we newly annotated hand-object tracks and contact states between hands and objects against videos in EPIC-KITCHENS dataset [28]. We collected tracks with various objects (*e.g.*, container, pan, knife, sink). The amount of the annotation was 1,200 tracks (67,000 frames) in total. We split the data into a training set (240 tracks), validation set (260 tracks), and test set (700 tracks) [1]. For the noisy dataset, we have generated 96,000 tracks with motion-based pseudo-labels.

[1] We extracted frames by either 30 or 25 fps, half of the original frame rate.

2.4.1 Implementation Details

We used FlowNet2 [57] for optical flow estimation. We used Adam [58] for optimization with a learning rate of 3e-4. We trained the network for 500,000 iterations with a batch size of one and selected the best model by frame accuracy on the validation set. The hyperparameters were set to $\delta_0 = 0.05, \delta_{end} = 0.25, \alpha = 0.01, \beta = 0.025, m = 2500$.

2.4.2 Evaluation Metrics

We prepared several metrics to evaluate the performance. **Frame Accuracy:** Frame-wise accuracy balanced by the ground truth label ratio; **Boundary Score:** F-measure of boundary detection. Performs bipartite graph matching between ground truth and predicted boundary [59]. Count as correct if the predicted boundary within six frames from the ground truth boundary; **Peripheral Accuracy:** Frame-wise accuracy within six frames from the ground truth boundary; **Edit Score:** Segmental metric using Levenshtein distance between segments [60]. We assume both contact and no-contact labels are foreground; **Correct Track Ratio:** The ratio of tracks which gives frame accuracy above 0.9 and boundary score of 1.0.

2.4.3 Baseline Methods

We compared our method against several baseline methods. **Fixed:** Predicts always as "contact"; **IoU:** Calculate the mask IoU between the input hand mask and object bounding box. If the score is larger than zero predicts as contact, otherwise no-contact; **ContactHands [27][2]:** Predicts as a contact if the detected hand's contact state is "object"; **Shan-Contact [22][3]:** Predicts as a contact if corresponding hand's contact state prediction is "portable"; **Shan-Bbox [22]:** Predicts as contact if there is enough overlap between the detected object bounding box and input object bounding box; **Shan-Full [22]:** Combines predictions of Shan-Contact and Shan-Bbox; **Super-**

[2] We used the pre-trained model provided by the authors along with their suggested hyperparameters.

[3] We used the pre-trained model provided by the authors, trained on 100DOH dataset and egocentric datasets.

Method	Frame Acc.	Boundary	Peripheral	Edit	Correct Ratio
Fixed	0.500	0.394	0.534	0.429	0.166
IoU	0.642	0.505	0.613	0.678	0.259
ContactHands [27]	0.555	0.440	0.596	0.468	0.136
Shan-Contact [22]	0.608	0.516	0.656	0.507	0.180
Shan-Bbox [22]	0.688	0.435	0.639	0.631	0.189
Shan-Full [22]	0.746	0.477	0.687	0.583	0.193
Supervised (train)	0.770	0.563	0.649	0.718	0.394
Supervised (train+val)	0.816	0.636	0.695	**0.793**	0.487
Proposed	**0.836**	**0.681**	**0.730**	**0.793**	**0.519**

Table 2.1: Results of hand contact state prediction performance.

Method	Frame Acc.	Boundary	Peripheral	Edit	Correct Ratio
Noisy Label only	0.780	0.569	0.703	0.687	0.344
Noisy + Trusted Label	0.811	0.624	0.708	0.759	0.453
Noisy + Trusted w/ PLC [47]	0.821	0.636	0.730	0.768	0.480
Pseudo-Labeling [61]	0.784	0.590	0.703	0.737	0.417
RGB	0.787	0.546	0.681	0.709	0.363
Flow	0.833	0.672	0.725	0.789	0.519
Proposed (RGB+Flow)	0.836	0.681	0.730	0.793	0.519

Table 2.2: Ablations on input modality and other robust learning methods.

vised: Our proposed prediction model, trained by trusted data alone. We note that for the **Shan-∗** baselines, the *100k+ego* pre-trained model provided by the authors was used, which is trained on egocentric video datasets including the EPIC-KITCHENS dataset.

2.4.4 Results

Quantitative results We report the performance in Table 2.1. Our proposed method consistently outperforms the baseline models on all the metrics, achieving a double correct track ratio compared to **IoU** based on the overlap between hand and object bounding boxes. The frame-based methods (**ContactHands**, **Shan-∗**) performed equal or worse than **IoU**, producing many false positive predictions. These results suggest that previous methods claiming contact state prediction fails to infer physical contact between hands and objects. While **Supervised** performed well, gPLC further boosted the performance by leveraging diverse motion-based cues with label correction, especially on boundary score.

Qualitative results Figure 2.6 shows the qualitative results. As shown in the top, our method distinguish contact and no-contact states by looking at the interaction between hands and objects while baseline methods yield false positive predictions by looking at box overlaps. The middle shows a typical no-contact case of a hand floating above an object. Our proposed model trained on motion-based pseudo-labels avoid producing false positive prediction.

Comparison against other robust learning methods To show the effectiveness of the proposed gPLC, we report ablations on other robust learning/semi-supervised learning methods (see Table 2.2 top). As expected, training using motion-based pseudo-labels performed worse due to labeling errors. Joint training with noisy and trusted labels gives marginal gain against the supervised model, but the boundary score remains low since it overfits against pseudo-label noise. We also applied the existing label correction method [47] on a single network with fine-tuning on trusted labels, but its performance was almost equal to joint training, suggesting that label correction on a single network does not yield good correction. We also tried a typical pseudo-labeling [61] without motion-based labels. However, it showed only a marginal improvement over the supervised baseline, suggesting that our motion-based pseudo-labels are necessary for better generalization.

Effect of input modality The bottom of Table 2.2 reports the ablation results of changing the input modalities. We observed that using RGB images alone impacts the boundary score, suggesting the difficulty of determining the contact state change without motion information. In contrast, the optical flow-based model achieved nearly the same performance as the full model, suggesting that motion information is crucial for accurate prediction.

Error analysis While our method can better predict contact states by utilizing the rich supervision from motion-based pseudo-labels, we observed several failure patterns. As shown in Figure 2.6 bottom, our method often ignored contacts when a person instantly touched objects without yielding apparent object motion. We also observed failures due to unfamiliar grasps, complex in-hand motion, and failure in determining object regions (see supplemental for more results). These errors indicate the limitation of the motion-based pseudo-labels which assigns labels only when clear joint motion is observed. To better deal with subtle/complex hand motions, additional supervision or rules on such patterns may be required.

How does gPLC correct noisy labels? To understand the behavior of gPLC, we measured how gPLC corrects labels during training. We included the validation set into the training data with two patterns of initial labels: (i) randomly corrupted labels from ground truth (with three different corruption ratios $c_r = 0.1/0.2/0.5$) (ii) motion-based pseudo-labels. We trained the full model and measured the accuracy of the labels for every epoch.

First, gPLC succeeded to correct randomly corrupted label even in the case of high corruption ratio of 0.5 (see Figure 2.7). However, in the case of a small corruption ratio of 0.1, gPLC made wrong corrections which means that both the noisy model and clean model got the prediction wrong. Improved boundary scores showed that gPLC can iteratively suppress inconsistent boundary errors. In the more realistic case of motion-based pseudo-labels, pseudo-labels were assigned to around 44% of the total frames, and achieved initial mean frame accuracy of 91.4% for the labeled frames. While gPLC reduced the error rate by 20% PLC wrongly flipped the contact state, which may have harmed the final performance (see Figure 2.8). These results indicate that gPLC effectively corrects noisy labels during training.

2.5 Conclusion

In this chapter, I have presented a simple yet effective video-based method of predicting the contact state between hands and objects, using appearance and motion information. I have introduced a semi-supervised framework of motion-based pseudo-label generation and guided progressive label correction that corrects noisy pseudo-labels guided by a small amount of trusted data. I have newly collected annotation for evaluation and showed the effectiveness of the proposed framework against several baseline methods. The model could be used to detect objects that are involved in interactions.

Algorithm 1 Guided Progressive Label Correction (gPLC)

```
Require: Noisy dataset $\tilde{S} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{N_p}$, trusted dataset $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_t}$, clean dataset $\hat{S} = \varnothing$, noisy model $f(\mathbf{x})$, clean model $g(\mathbf{x})$, initial and end thresholds $(\delta_0, \delta_{end})$, correction threshold $\delta = \delta_0$, flip ratio $\alpha$, step size $\beta$, supervision interval $m$, total round $N$
Ensure: Trained Model $f(\mathbf{x})$
  $\hat{S} = \{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}_{i=1}^{N_p}$ where $\hat{\mathbf{y}}_i$ is a list with empty elements same size as $\tilde{\mathbf{y}}_i$    // Initialize clean dataset
  for $n \leftarrow 1, \dots, N$ do
    for $i \leftarrow 1, \dots, N_p$ do
      $\mathbf{z}_i \leftarrow \tilde{\mathbf{y}}_i$    // Keep previous labels
      for $t \leftarrow 1, \dots, |\tilde{\mathbf{y}}_i|$ do
        if $\tilde{y}_i^t \in \{0, 1\}$ and $|f(\mathbf{x}_i^t) - \frac{1}{2}| \geq \frac{1}{2} - \delta$ and $\mathbb{I}_{\{f(\mathbf{x}_i^t) \geq \frac{1}{2}\}} = \mathbb{I}_{\{g(\mathbf{x}_i^t) \geq \frac{1}{2}\}}$ then
          $\tilde{y}_i^t, \hat{y}_i^t \leftarrow \mathbb{I}_{\{f(\mathbf{x}_i^t) \geq \frac{1}{2}\}}$    // Refine or add label by confident prediction
        end if
      end for
      Train $f(\mathbf{x})$ on $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$ and $g(\mathbf{x})$ on $(\mathbf{x}_i, \hat{\mathbf{y}}_i)$    // Update models
      if #iterations % $m = 0$ then
        Train $f(\mathbf{x})$ and $g(\mathbf{x})$ on $S$    // Fine-tune on trusted set
      end if
    end for
    if $\sum_{i,t} \mathbb{I}_{\{\tilde{y}_i^t = z_i^t\}}$ ¡ $\alpha \cdot \sum_{i,t} \mathbb{I}_{\{\tilde{y}_i^t \in \{0,1\}\}}$ then
      $\delta \leftarrow min(\delta + \beta, \delta_{end})$    // Loosen threshold if number of flipped labels are small enough
    end if
  end for
```

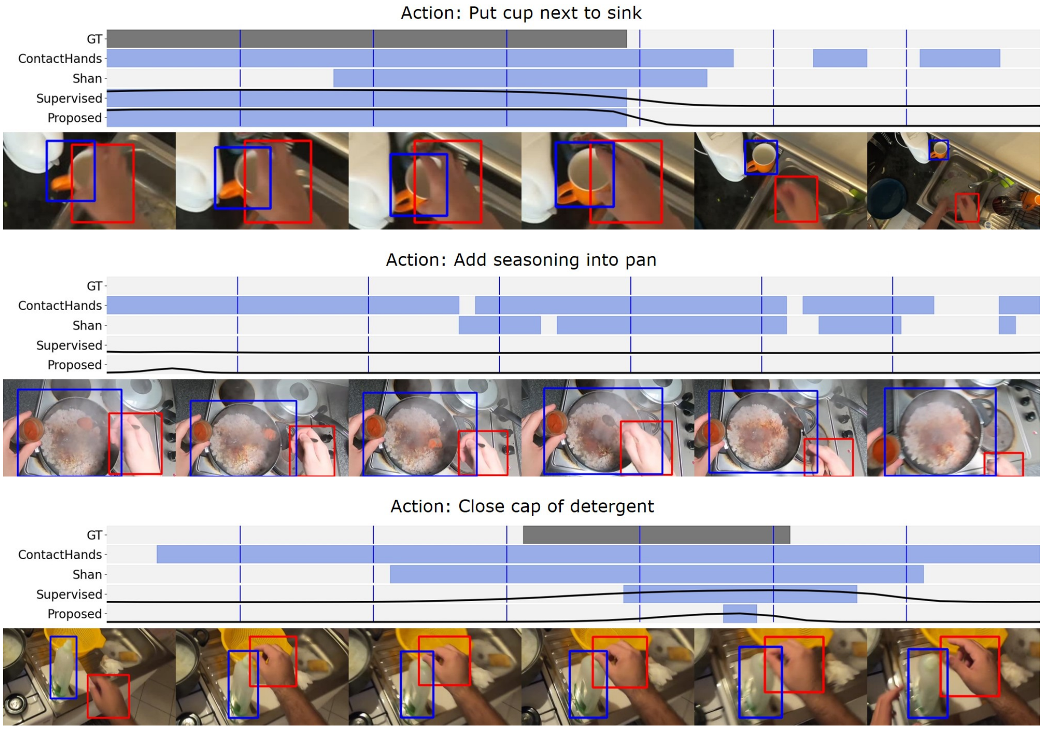


Figure 2.6: Qualitative examples. Upper chart shows ground truth contact state and prediction of each model (gray and blue region indicates contact, otherwise no-contact) with contact probability in black line. Lower images correspond to blue vertical lines in chart from left to right and red and blue boxes represents input hand and object bounding box.

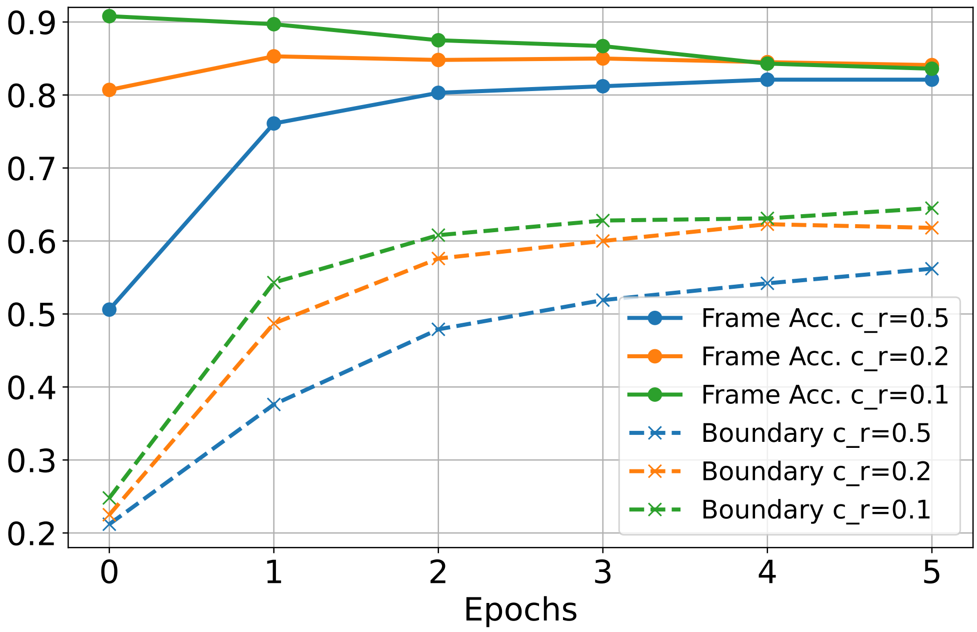


Figure 2.7: Accuracy of noisy labels when initialized by corrupted ground-truth labels. Horizontal axis shows elapsed epochs ("0" denotes initial labels). Vertical axis shows frame accuracy (solid) and boundary score (dashed).

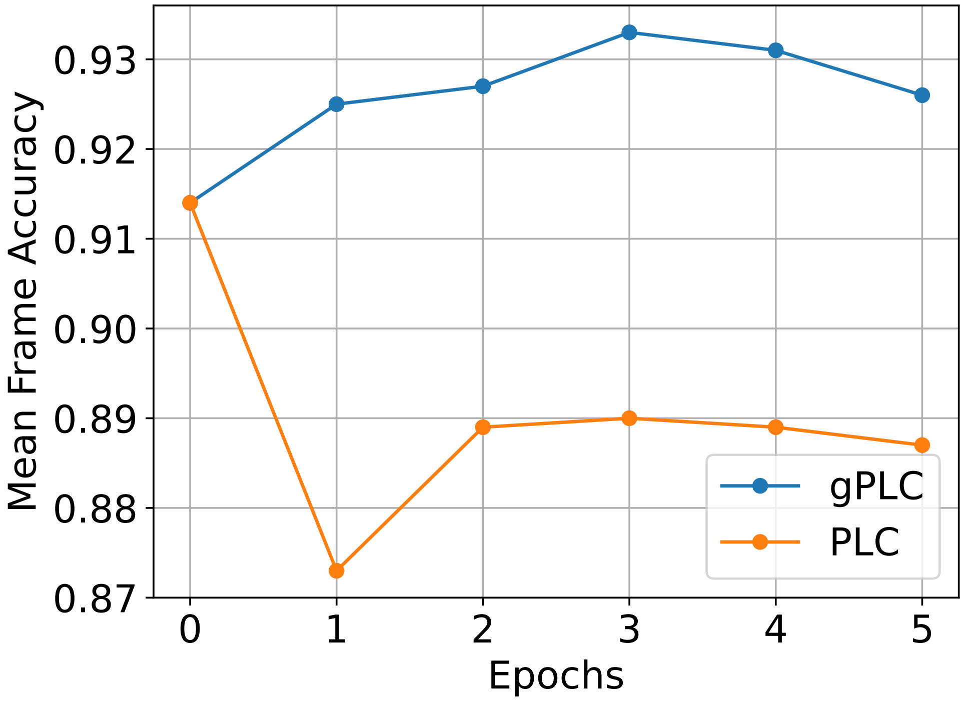


Figure 2.8: Accuracy of noisy labels when initialized by motion-based pseudo-labels. Horizontal axis shows elapsed epochs and vertial axis shows mean frame accuracy per track. Note that non-labeled frames are ignored.

Chapter 3

Category-Agnostic Object Instance Identification

3.1 Introduction

Object instance identification is a long-standing but yet unsolved problem in computer vision. We humans can recognize that a specific object is different from other objects. For example, we typically have a specific mug or smartphone we use every day and distinguish them from other objects. We remember the specific buildings around the neighborhood, know where all the kitchen tools are placed in the kitchen, or find out friends from a large crowd. The ability to visually identify unique objects is essential for intelligent agents to continuously collaborate with humans in a shared space. By recognizing that an object is the same as the one used before, a robot can plan for a longer period of time and respond to the user's request conditioned on their context (*e.g.*, find my phone and bring it to me). The problem has many practical applications such as surveillance [62], product search [63], dexterous manipulation [64], and assistive technology [65, 66].

Over the last two decades, category-level object recognition, which classifies objects into specific categories, has been primarily studied. On the other hand, instance-level object identification, which discriminates specific objects appearing in a scene, also offers unique challenges not included in category-level recognition. First, it does not assume any pre-defined categories in na-

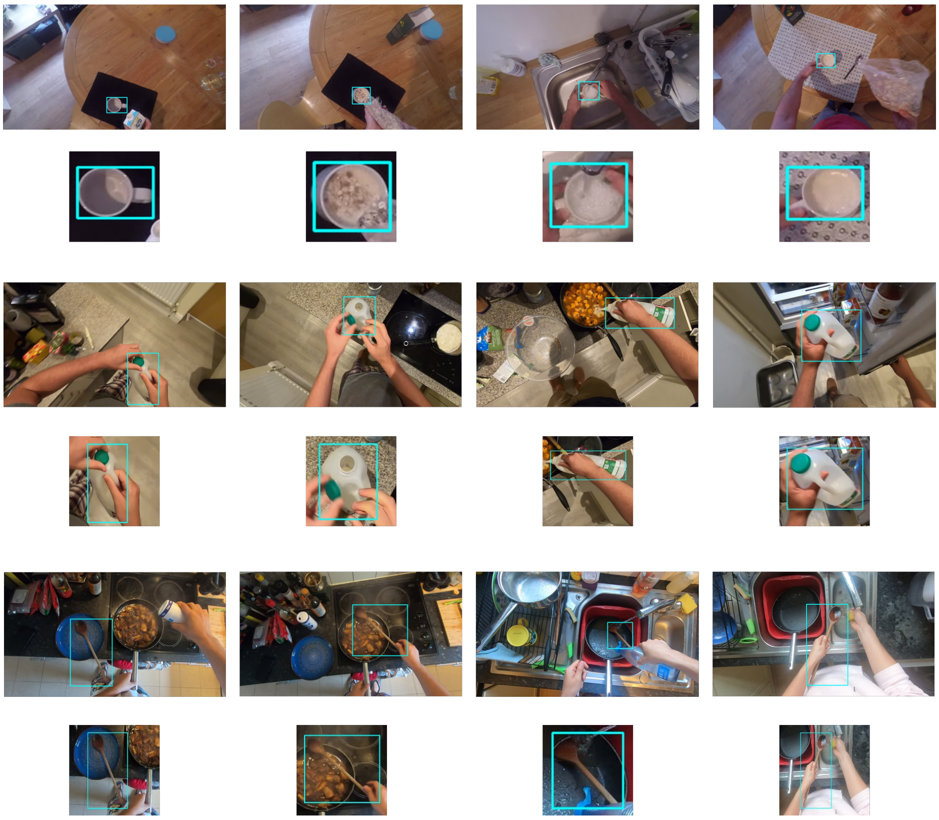


Figure 3.1: Example images of EK-Instance Dataset. Each row represents images of same instance.

ture. Arbitrary types of instances appear and the model has to differentiate all of them. Second, the model must distinguish the fine-grained differences between objects from similar categories. For example, similar-looking mugs may exist but each of them usually has distinctive differences among each other. Beyond classifying an image into a category, the model must correctly distinguish them by looking at their details. Thirdly, instance-level recognition requires generalization against unknown instances. In category-level recognition, object categories are usually shared among training and testing samples. However, instances are usually unique across environments, and it is unrealistic to collect training samples for all of them.

While object instance identification in general has been studied extensively in the early days [67–70], recent works have been focusing on category-specific settings such as recognizing specific person [62], vehicle [71], landmarks [72], artwork [73] or faces [74]. Despite its importance, object instance identification in a category-agnostic setting has not been investigated in detail. Previous works on category-agnostic object instance recognition were typically studied in a controlled environment, aiming to learn a classifier robust against changes in object pose, viewpoint, and illumination. They assume objects to be distinctively captured apart from other objects or backgrounds. Moreover, objects are assumed to be immutable—implicitly limiting the range of instances to be recognized.

In this chapter, the problem of category-agnostic object instance recognition in a real-world setting is studied. In the real world, each object will be used by the user for its purpose. Users actively interact with multiple objects, for example, cutting vegetables on a cutting board with a knife. Objects appear as a constituent of a scene rather than a unit. During the user's activity, an object will undergo a significant appearance change to accomplish its job. For example, a mug is used as a container for pouring coffee; food and cutlery are placed on the plates. Manipulation by the hands may produce abrupt motion and severe occlusion. In such a situation, the immutable assumption breaks down and the intra-class variance of an instance will be much larger than that in an ideal environment. The environment also changes over time, causing changes in object pose and lighting condition on the reappearance. I aim to build a general model of identifying unknown object instances in such in-the-wild situations.

First-person videos are selected as a subject that reflects the above difficulties. First-person videos can capture long and continuous activities over time up-close. Therefore, it is suitable to capture the natural interaction and appearance changes of objects appearing in daily life. To this end, I propose the EK-Instance dataset, a challenging benchmark built on the large-scale EPIC-KITCHENS-100 dataset [23] that contains daily cooking activities across different time and environments. I collect 1,554 object instances from 23 participants with around 90,000 frames, which could be used to train a modern neural network-based model. Figure 3.1 shows some examples of the annotated images. As seen in the figure, the object's appearance significantly changes by the action made by the camera wearer. Such actions produce significant foreground change (first row), severe occlusion (second row), and interplay between other objects (third row), which makes the problem more challenging. The dataset contains a significantly larger number of instances compared to the existing datasets (from tens to 400 instances), which enables cross-instance evaluation. Since the dataset is collected from unscripted first-person videos, the frequency of appearance among instances are significantly imbalanced which makes further difference among the existing datasets where the number of samples are balanced among instances.

On annotation, I find some cases that are difficult to judge whether the images are from the same instance or not because objects are often combined or separated during an interaction, Therefore, I establish several principles to produce consistent annotations that could be solved by looking at appearance.

As I aim to learn a general model of identifying unknown object instances, I formulate the problem as a clustering problem. Compared to the re-identification setting, Query object images are not assumed to be given. Specifically, the problem is formulated as clustering of object instance tracks each composed of object image tracks. Strong baseline models using off-the-shelf metric learning algorithms and clustering algorithms are introduced. Based on the above datasets and models, extensive analyses are performed to extract insights needed to perform object identification in a dynamic environment. Specifically, (i) an analysis on cross-dataset transfer and (ii) visualization of learned representation are performed to assess the generalizability and limitations of learned models.

The contributions of this work are as follows. First, I build a new EK-Instance dataset for object identification which includes more than 1,500 objects taken from unconstrained first-person videos. Second, I show that the models trained by the EK-Instance dataset show better robustness against both foreground and background clutter. They show robustness against hand occlusion, significant appearance changes, and background changes upon movement. The cross-dataset evaluation reveals that the datasets collected in a controlled environment do not generalize to the real-world setting while the model trained with the EK-Instance dataset showed better cross-dataset transferability. Finally, we show notable failure patterns to be solved, along with suggestions for improvement. We insist that (i) robustness against object state-change (ii) Integration of low-level and high-level concepts (iii) an active mechanism to determine the foreground object are required for further development.

3.2 Related Work

3.2.1 Instance Recognition

Instance recognition is formulated to recognize a particular object of the scene from an image instead of its category. Recognition of object instances was mainly studied in the context of robotics to make the robot recognize object instances to be grasped [64, 75–77]. Not only classifying the object images but object instance detection from unlabeled images is also studied as an important topic [78–82]. Mainly focusing on robotics application, it was basically studied in a controlled, small-scale setting, where the number of objects is at most several hundred.

While instance recognition methods are defined to recognize *a particular object*, the definition of an object instance differs across methods and datasets. In the context of robotics, rigid household objects such as bottles, cups, cellphones, books are generally used as a subject. Assuming that the target scene is static and the spatial configuration does not change over time, some works try to discriminate visually identical instances [83, 84]. This group also assumes that the object's shape or appearance does not change across moments. Robustness against different poses, viewpoints, and background is their primary interest. In the context of product recognition, products with the same model are considered to be the same instance even if their entities are different (*e.g.*, [85]). This assumption is reasonable when the appearance of a same product is visually indistinguishable. This situation also happens in the daily living domain because for example we often use a set of visually identical dishes or glasses interchangeably. In a dynamic scene where the spatial configuration of objects changes across time, it will be often difficult to precisely identify a particular object when visually identical entities exist. Therefore, we adopt the latter assumption by limiting the scope to be solved within the range of vision-based techniques, which will be further discussed at Section 3.3.

To recognize instances, traditional methods used low-level features such as color histogram or local descriptors [78, 79] followed by keypoint matching such as RANSAC. However, deep neural networks are shown to be effective in recent works [75, 76]. Especially, metric learning are used to learn discriminative feature of objects from a small number of examples [64, 77].

It is worth mentioning the differences in problem settings. Most works assume that the training images of the test object instance are given [75, 77, 86], 3D models of the target instance given [81, 82], or target objects explicitly given as query images [64, 83, 84]. However, the above assumptions require knowledge of the target object instances to be recognized, which is cumbersome and does not scale to unknown instances. Instead of training a particular object classifier using explicit training examples, we aim to build a model which can discover the concept of instances from training data apart from testing instances. To this end, we formulate the problem as clustering of unlabeled images into a group of object instances.

3.2.2 Instance Re-Identification

Along with recognition, instance re-identification tasks are studied mostly in category-specific settings (*e.g.*, person [62], vehicle [71], landmark [72], artwork [73], and face [74]) in the context of surveillance and retrieval. In this setting, query images are given in advance and asked to find the same instance from the testing images. Person/vehicle re-identification showed that metric learning is effective to discriminate different identities by learning a feature space where samples from the same identity are close together while samples of different identities are far apart [87]. By utilizing domain knowledge such as body parts, part-based model [88], viewpoint-specific representation [89], and relation between parts [90] were proved to be effective. However, in our category-agnostic setting, it is unrealistic to assume a fixed structure to arbitrary types of objects. Therefore, a more general solution that does not require knowledge on object structure is demanded.

3.2.3 Instance-Level Dataset

Along with applications such as product recognition, scene understanding, 3D reconstruction, and robot manipulation, numbers of instance-level datasets have been proposed. Their source are roughly divided into (i) controlled environments [86, 91–95], (ii) crawling from the Internet [85, 96], and (iii) crowdsourcing [97, 98].

Primarily focusing on robotics application, the first group collects data in mostly controlled settings such as turntable [92, 93] or monotonous background [94, 95, 99]. Robustness against rotation, translation is their primary interests but they are typically collected in a small number of environments with less background clutter. Although few works included real-world samples, they did not include instances during natural interaction. For example, the INSTRE dataset [100] consists of 200 object instances that are a combination of the Internet images and manually-captured images. Manually captured images are taken at various places, which offers significant background variability. However, the captured images are intentionally separated from the user's context and do not reflect the difficulties in the actual living space.

A notable exception is [91] which study the problem of recognizing hand-held objects captured in first-person videos. While they are aware of the difficulties in recognizing hand-held objects in the early days, their dataset only contained 42 objects, which is not sufficient to train a modern neural network-based model. Furthermore, each sequence was captured in a scripted manner, which hid the challenges that appears in real-world activities. The common problem of the datasets belonging to this group is that the number of instances is limited at most to a few hundred, which is not enough for training a general object identification model which generalizes to unknown instances.

The second group collects data from large E-Commerce sites for the product recognition task. Compared to the first group, it is easier to scale up the number of instances. Therefore, datasets with more than 10k instances are proposed in contrast to the first group. For example, Song *et al.* [85] collected 120k images from 22,634 online product categories in eBay. Although it contains diverse viewpoints due to its original purpose, it tends to be only captured in a single environment, and most of the products are shot alone, separated from other disturbances. This makes the dataset vulnerable to background/foreground clutter, which will be later verified in experiments.

The last group collects data for data-driven 3D reconstruction. To enable category-centric 3D reconstruction from a limited number of viewpoints, crowdsourcing are used to collect a vast amount of multi-view images and its 3D point clouds. For example, the Common Objects in 3D (CO3D) dataset comprises of around 19,000 videos capturing 50 MS-COCO categories. Although the size and quality of images are useful for large-scale training, objects do not undergo any state change or manipulation nor include significant foreground and background clutters.

Different from the above datasets, we collect a large number of object instances that appear in real-world environments, including objects under natural manipulation action and significant foreground/background clutters. We choose a large-scale, unscripted egocentric video database as a source. The obtained annotations provide unique challenges on learning an object instance identification model which generalizes to unknown and unconstrained scenes.

3.2.4 Instance-Level Image Clustering

Unsupervised image clustering has been also studied as a long-standing problem in computer vision and has been used for discovering object instances in a bottom-up manner. Specifically, clustering of faces [101–109] and landmarks [72] has been studied based on applications such as face recognition and image retrieval. Because these fields process a significantly large number of clusters (*e.g.*, 100k), not only accuracy but also computational cost are considered as important aspects. While general clustering methods such as K-Means [110], Spectral Clustering (SC) [111], Hierarchical Agglomerative Clustering (HAC) [112, 113], and Approximate Rand-Order (ARO) [101] works well when optimal data density distribution is obtained through representation learning, they do not capture the contextual, higher-order relationship between data points on performing clustering. The use of hard constraints [102] alleviate the problem but still, they have limitation in representational power.

In the above context, recent work aims to learn the underlying structure of facial/landmark images in a data-driven manner. To learn higher-order relationships beyond pairwise relation, Graph Neural Networks [103, 104, 106] and Transformers [105] are shown to be effective. Also, discovering high-density (high-confidence) samples are found to be effective in linking face images [104, 106, 114].

However, applying the above methods are far more challenging because of the class-agnostic assumption, larger intra-class variation, and imbalanced distribution. Because arbitrary types of objects appear, a typical view of an object will not exist. Environmental clutters create larger intra-class variations which produce multiple density peaks within a single instance. Furthermore, the imbalanced appearance across instances makes it difficult to determine an appropriate threshold to predict the connection between data points. Therefore, we resort to traditional methods rather than applying the modern unstable methods to focus on finding the difficulties in solving this task.

3.3 EK-Instance Dataset

In this section, we introduce the EK-Instance dataset which offers a challenging benchmark on object instance identification in a dynamic and cluttered environment (see Figure 3.1 for example). The annotation is built on natural interactions of multiple people and includes significant foreground and background clutters which was not present in the existing datasets.

3.3.1 Data Source

The annotation is made on the EPIC-KITCHENS-100 [23] dataset which contains 100 hours of natural kitchen activities captured from a head-mounted wearable camera. We choose first-person videos as a subject because it captures continuous activities over time. Some habitual activities (*e.g.*, preparing a cereal for breakfast) are repeated in different scenes, posing a natural challenge in finding re-appearing object instances. 37 participants recorded activities in different kitchens, which gives variation in environments, objects, and activities.

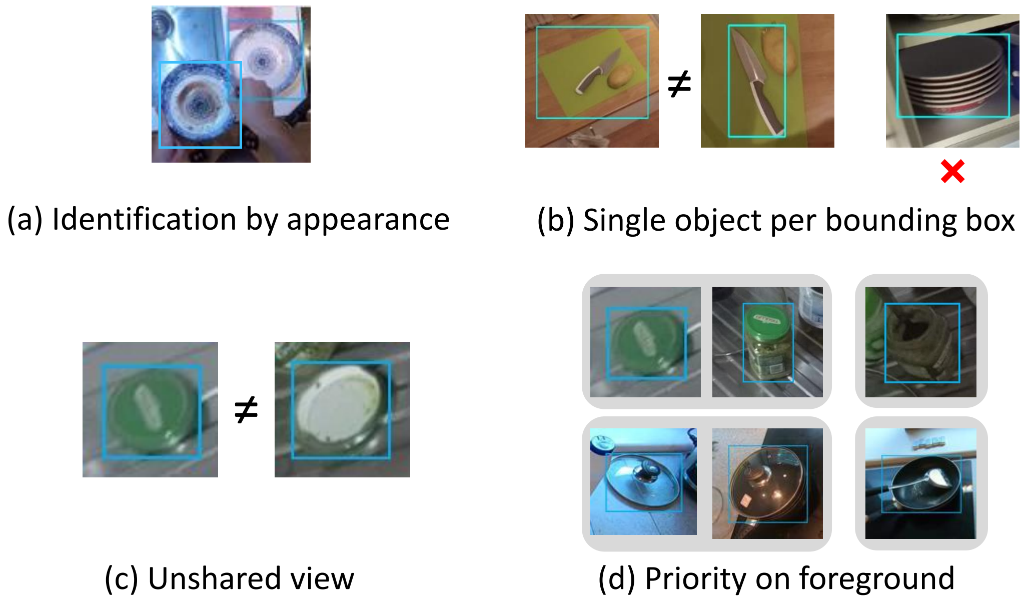
3.3.2 Definition of Instance

Before introducing the annotation procedure, we clarify what *an instance* refers to. Compared to the previous works that captured objects apart from clutters, real-world videos include many corner cases caused by the interplay between background and other objects. In addition, external knowledge beyond appearance is often required to accurately identify the same instance appearing in a different scene. For example, tableware typically includes a set of glasses or dishes with the same look. Food or commercial products will be consumed and disposed of. Thus, one appearing in a different scene may not be the same instance that has been seen before. Distinguishing such visually identical instances is impossible and will be beyond the focus of computer vision.

To provide well-defined and consistent annotation, we propose four principles to be observed. Figure 3.2 shows examples of the principles we made.

(a) Identification by appearance: First, we define an instance as *an object which has a unique appearance that is distinguishable from other objects.* Therefore, visually identical entities will be counted as a single instance. For example, if there is a set of visually identical tableware, we consider them as a single instance.

(b) Single object per bounding box: Secondly, we use bounding boxes on determining the target object. If multiple objects are intersecting with each other, we specify each of them by the bounding box that surrounds the object just enough. Figure 3.3 shows a schematic example of determining a target object from a bounding box. If an object uniquely fits with a bounding box, we determine it as valid and include it in the dataset. However, if there are more than or equal to two objects fit by a bounding box, we determine them as invalid and remove them from the dataset. Also, we consider as



(a) Identification by appearance

(b) Single object per bounding box

(c) Unshared view

(d) Priority on foreground

Figure 3.2: General principles on annotation. Rectangle denotes target bounding box.

invalid if multiple objects are surrounded by a bounding box (Figure 3.2 (b)).

(c) Exception on unshared view: Ideally, a model should be viewpoint-invariant and recognize instances even its viewpoints are different. Therefore, in principle, we consider images as the same instance even the views are not shared among images. However, there exist cases when there is no clue on predicting a view from the other view without external knowledge. we allow an exception in such situations. For example, in the case of Figure 3.2 (c), the color of the lid upside down is completely different so determined to be a different instance.

(d) Priority on foreground: Some objects can be separated into multiple parts. For example, a frying pan set can be separated into a body and a lid. If we look at it from a top-down view, it will be impossible to distinguish between (i) a set of body and lid and (ii) the lid. While multiple interpretations are possible, we give priority to the foreground and count as same if

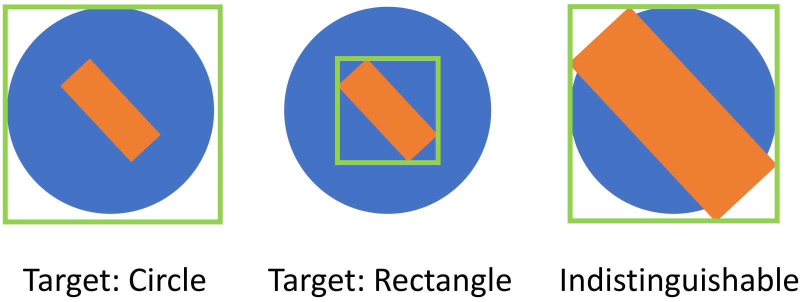


Figure 3.3: Schematic example on determining the target object.

objects share the same foreground. In the case of Figure 3.2, we consider (i) and (ii) as the same while considering the body alone as a different instance.

These principles help the annotators to provide consistent annotations among irregular cases that occur in real-world environments.

3.3.3 Annotation Procedure

Based on the above principles, we annotate object bounding boxes and the correspondence between them. Concretely, 2D bounding boxes which surround the object instances are annotated to the video frames and they are associated by marking whether a bounding box pair is from the same instance or not. However, it turned out to be labor-intensive to thoroughly annotate all the potential object pairs. Therefore, we adopt a semi-automatic procedure to reduce the burden.

First, we apply a class-agnostic object detector [115] to obtain an object-like region across videos. Next, we apply a motion-based object tracker [52] to form a short track of bounding boxes. While this procedure produces a large number of object tracks, they also include numerous irrelevant detections such as body parts (*e.g.*, hand, foot, and arm), static structure (*e.g.*, faucet, cooking stove, kitchen sink, and furniture), non-object (*e.g.*, shadow, part of object, set of multiple objects, and background), and low-quality detections (*e.g.*, blurred). These irrelevant detections are manually filtered out and not further processed. Also, detections around the image boundary or with severe occlusion (when more than 50% of the original object area is occluded)

Split	#participants	#instances	#tracks	#frames
Training	16	1051	27234	64300
Validation	3	193	4647	11851
Test	4	310	6598	16225
Total	23	1554	38479	92376

Table 3.1: Dataset statistics.

are removed. After the filtering, the remaining object tracks appearing at different times are associated with each other by the annotator if they are from the same object instance. After the track association, we sampled image detections every one second from each track. Because these detections do not necessarily cover the entire object, we corrected the bounding boxes so that each bounding box covers the object accurately. This semi-automatic procedure allowed us to produce annotation around 20 hours of video, which is 1/5 of the original dataset length.

3.3.4 Dataset Statistics

As a result, we collected 1,554 object instances, 38,479 tracks and 92,376 frames from 23 participants in total. The instances are collected from diverse categories. The majority consists of container (*e.g.*, storage container, cup, mug, glass, bottle, box, and seasoning), cooking tools (*e.g.*, frying pan), tableware (*e.g.*, knife, fork, spoon, spatula, and tongue), food (*e.g.*, apple, banana, and bread), electronics (*e.g.*, toaster, coffee machine, mixer), and other kitchen tools (*e.g.*, paper roll, sponge, detergent, and towel). However, the object categories are not limited to the above, including objects which is difficult to define a clear category (*e.g.*, fridge magnet, portafilter, and package of specific products). Based on the recommended split in the original benchmark, the data was divided into three splits. Specifically, P33, P34, and P36 were used for validation, and P09, P11, P18, and P32 were used for testing. Table 3.1 summarizes the statistics of each split. Since the annotations were collected from multiple participants, cross-subject evaluation is possible while keeping enough training data for training.

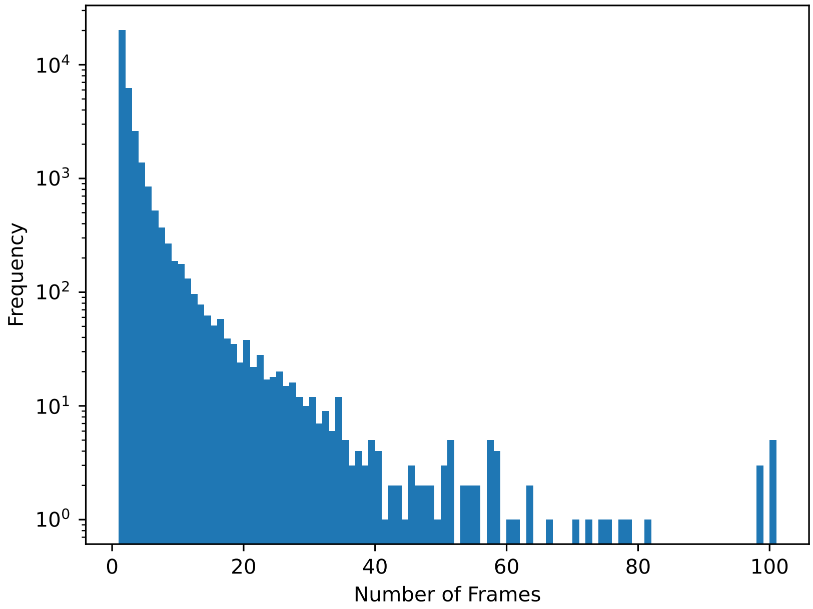


Figure 3.4: Distribution of number of frames per track.

Since the data is collected from real-world videos, it reflects the imbalanced frequency of appearance. Figure 3.4 and 3.5 summarize the number of frame per track and the number of track per instance, respectively. While the majority of the instances appear a few times for a short duration, a small number of instances appear frequently.

3.4 Method

To evaluate the task of object instance identification, we introduce strong baseline models based on metric learning.

3.4.1 Problem Statement

The problem is formulated as the clustering of image tracks to a group of instances. First, we assume arbitrary types of objects detected from an object detector. We also assume that short-term object tracks are obtained

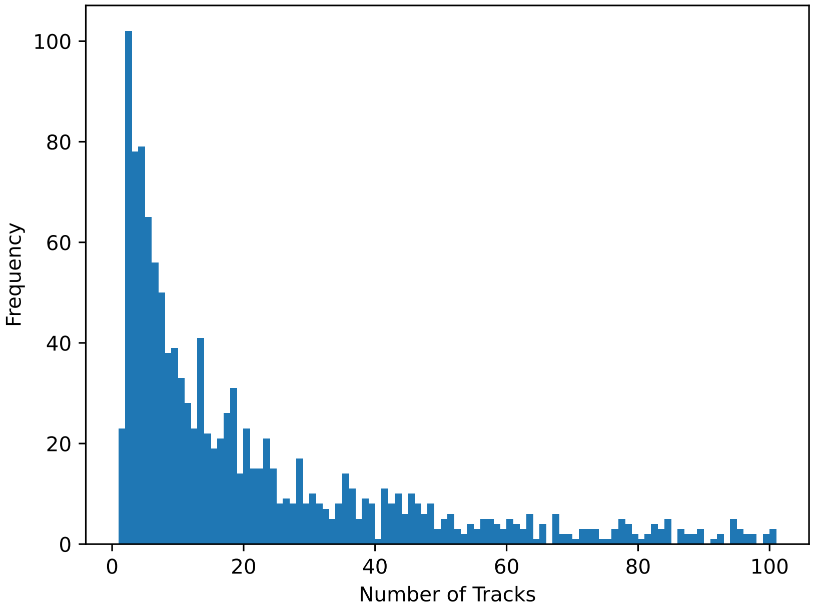


Figure 3.5: Distribution of number of tracks per instance.

by applying an object tracker to them. Given a set of detected image tracks $\{X_1, \ldots, X_N\}$, we want to assign unique cluster labels $\{y_1, \ldots, y_N\}$ ($y_i \in \{1, \ldots, K\}$), which is grouped by instance. It is reasonable to apply tracking before clustering because we can efficiently aggregate similar feature vectors. Also, tracking helps the model to acquire robustness against different poses or viewpoints.

3.4.2 Image Track Encoder

Given a image track $X_i = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ ($\mathbf{x_j}$ denotes an object image), we want to obtain a low-dimensional encoding $\mathbf{z}_i$ which will be passed to a clustering algorithm. Throughout this study, we use ResNet-34 [116] pre-trained with the ImageNet dataset [117] as a backbone network to extract frame-level feature. We pass the images to the backbone layers before the final average pooling layer and obtain a 512-dimensional feature vector for each image. Next, they are further averaged between tracks to aggregate image-level features to a track-level feature vector. Finally, we pass it to a single fully-connected layer to obtain a 256-dimensional track embedding.

3.4.3 Training

We employ three metric learning loss functions to train the image track encoder.

Normalized softmax (N-Softmax) loss The most straightforward form of metric learning by classification is the normalized softmax loss [118], which is a cross entropy loss but the embeddings and the weight vector W are L2-normalized.

$$L_{\text{nsoftmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(W_{y_i}^{\top}\mathbf{z}_i'\right)}{\sum_{k=1}^{K}\exp\left(W_k^{\top}\mathbf{z}_i'\right)},$$

where $\mathbf{z}_i' = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}$ is a L2-normalized vector of $\mathbf{z}_i$ and τ is a temperature parameter to balance between positive logits and negative logits.

Additive angular margin (ArcFace) loss The ArcFace loss [119] is represented as

$$L_{\text{arcface}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(\cos\left(\theta_{y_i}+m\right)/\tau\right)}{\exp\left(\cos\left(\theta_{y_i}+m\right)/\tau\right)+\sum_{k=1,k\neq y_i}^{K}\exp\left(\cos\theta_j/\tau\right)},$$

where θ_k is the angle between the L2-normalized weights W_k and the input feature $\mathbf{z}_i'$ belonging to the y_i-th class, and m is the angular margin penalty to increase the inter-class distance. Given training instances, this loss tries to learn a embedding to have enough margin m from other instances, achieving a compact representation for each instance.

N-pair loss The n-pair loss [120] aims to directly increase the similarity of positive pairs (features from a same class) while decreasing the similarity of negative pairs (features from different classes) using the softmax function. This loss learns pairwise relation efficiently by taking all the pairs within the batch and does not require a class prototype weight vector W on training.

$$L_{\text{npair}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\sum_{j=1,y_j=y_i}^{N}\exp\left(\mathbf{z}_i^{\top}\mathbf{z}_j/\tau\right)}{\sum_{j=1}^{N}\exp\left(\mathbf{z}_i^{\top}\mathbf{z}_j/\tau\right)}.$$

3.4.4 Inference and Clustering

At inference, we extract the 256-dimensional track feature for each track and apply clustering on those feature vectors. Given trained feature embeddings, we evaluate two well-known clustering algorithms: Spherical K-means [121] and HAC [112]. The spherical K-means algorithm takes the L2-normalized vector as input and applies clustering in a spherical space by normalizing the centroid vectors on each iteration. We use cosine similarity for similarity measures in both algorithms. On using K-means, we assume that the number of instances is known and set the number of clusters equal to the ground truth. In HAC, we finetuned the hyperparameter by the AMI score (introduced later) on the validation set.

3.5 Experiments

3.5.1 Implementation Details

All the images are resized to 224×224. We used Adam [58] for optimization with a learning rate of 3e-5. We set the batch size to 448, uniformly sampling 112 instances $\times$ 4 tracks per batch. Starting with the backbone network pre-trained by the ImageNet dataset [117], we fine-tuned the network for at most 4,000 iterations with a batch size of one and selected the best model using the validation set. The training was done within an hour using a single NVIDIA A100 GPU.

3.5.2 Evaluation Protocol

Data Split We evaluate the model on a per-participant basis. In particular, given a split with multiple participants, we evaluate the clustering performance for each participant and report their average score.

Metrics We employed four metrics popularly used for evaluating clustering performance.

Adjusted Mutual Information (AMI) [122]: A metric based on mutual information. Returns a value of 1 when the two partitions are identical.

Given a ground truth cluster set $C = \{c_i\}_{i=1}^{N}$ and a predicted cluster set $K = \{k_i\}_{i=1}^{N}$, AMI is defined by:

$$AMI = \frac{I(C,K) - \mathbb{E}[I(C,K)]}{\frac{1}{2}[H(C) + H(K)] - \mathbb{E}[I(C,K)]},$$

where I and H denotes the mutual information and the entropy, respectively.

Unsupervised clustering accuracy (ACC): The maximum accuracy achieved by searching over all one-to-one mappings between clusters and labels. Hungarian algorithm [123] are used to find the optimal assignment m:

$$ACC = \max_{m} \frac{\sum_{i=1}^{N} \mathbf{1}\{c_i = m(k_i)\}}{N}.$$

Paired F-score (F_P): Given N tracks, we take all the $\frac{1}{2}N(N-1)$ pairs in the evaluation set and calculate the precision and recall of them, counting as correct if the relation (in the same cluster or not) of the pair by the predicted clusters is the same as that by the ground truth classes.

BCubed F-measure (F_B) [124]: This measure defines cluster-level precision. Given $L(i)$ and $C(i)$ denotes the predicted cluster and the ground truth cluster, respectively, the correctness between two points i and j are defined as:

$$Correctness\ (i,j) = \begin{cases} 1 & \text{if } L(i) \ = \ L(j) \text{ and } C(i) \ = \ C(j) \\ 0 & \text{otherwise.} \end{cases}$$

The precision and recall are defined as:

$$BCubed\ Precision = \frac{1}{N} \sum_{i}^{N} \sum_{j \in C(i)} \frac{Correctness\ (i,j)}{|C(i)|},$$

$$BCubed\ Recall = \frac{1}{N} \sum_{i}^{N} \sum_{j \in L(i)} \frac{Correctness\ (i,j)}{|L(i)|}.$$

Naïve baseline In addition to the three metric learning models, we also evaluated the ImageNet pre-trained model without fine-tuning. Specifically, we used the 512-dimensional embedding just before the last layer. We note that the ImageNet pre-trained model is already a strong baseline model because it is trained by diverse categories of images with various backgrounds, trying to find discriminative features useful for category-label classification.

Model	Clustering Method	AMI	ACC	F_P	F_B
ImageNet	Spherical K-means	0.755	0.590	0.547	0.628
	HAC	0.849	0.735	0.724	0.770
N-Softmax	Spherical K-means	0.803	0.631	0.606	0.684
	HAC	0.892	0.799	0.789	0.828
ArcFace	Spherical K-Means	0.794	0.631	0.606	0.670
	HAC	0.892	0.796	0.784	0.827
N-pair	Spherical K-Means	0.834	0.713	0.646	0.724
	HAC	**0.924**	**0.854**	**0.847**	**0.874**

Table 3.2: Results of clustering results on EK-Instance dataset.

3.5.3 Quantitative Results

Table 3.2 shows the results on EK-Instance dataset. While the ImageNet pre-trained model showed moderate performance, the models fine-tuned by the training set showed significant improvement. Specifically, the combination of N-Pair loss and HAC gave the best performance. Although the true number of classes was not given, HAC achieved better performance because K-means assumes each cluster to have a uniform density but the actual distribution was highly imbalanced.

3.5.4 Qualitative Results

Successful cases Figure 3.6 shows successful examples of the best performed model (N-pair + HAC). As shown in the top row, the model was able to capture the variation in pose and occlusion by hand. We also observed robustness against significant foreground clutter (second row). While ingredients are cooked on the frying pan, the model was able to absorb the appearance change by looking at the peripheral regions. We note that the bottom two rows of the second example are considered as a different instance following the foreground-priority principle, although the body was visible through the transparent lid. The third example also showed strong robustness against appearance change, which shows the unique aspect of the EK-Dataset.

Dataset	EK-Instance			Online Products			INSTRE			CORe50		
	AMI	F_P	F_B	AMI	F_P	F_B	AMI	F_P	F_B	AMI	F_P	F_B
ImageNet	0.849	0.724	0.770	0.332	0.023	0.434	0.944	0.836	0.890	0.497	0.311	0.446
EK-Instance	**0.924**	**0.847**	**0.874**	0.358	**0.360**	0.448	0.930	0.837	0.884	0.578	0.415	0.532
Online Products	0.800	0.620	0.694	**0.649**	0.321	**0.633**	0.870	0.756	0.773	0.090	0.063	0.216
INSTRE	0.795	0.634	0.703	0.304	0.047	0.445	**0.984**	**0.963**	**0.972**	**0.667**	**0.440**	**0.599**

Table 3.3: Results of cross-dataset transfer. Left column shows source dataset and top row shows target dataset. Numbers in bold and blue text show best and second best results across source datasets, respectively.

Failure cases Since HAC performs bottom-up iterative merging, we visualized moments when tracks from different instances are merged. Figure 3.7 shows failure cases on the best-performed model. Although the model showed high performance in general (0.847 in pairwise F-score), we have found several distinctive failure patterns. **(i) Intra-category confusion**: First, the model struggled to spot fine-grained differences in the same category (first row, first and second column). While we found differences in color and unique symbols, the model failed to distinguish such differences. **(ii) Semantic confusion**: Secondly, we found some false merging between semantically different objects (first row, third and fourth column). They showed high similarity when the color patterns were similar to each other. **(iii) Foreground confusion**: This error reflects the notable difficulty specific in a real-world setting (second row). While the same food is appearing in most of the tracks, its containers are different. **(iv) Missed target object**: This error is closely related to the bounding box priority principle (third row). In the first example, a knife on a chopping board and the chopping board are wrongly merged as a single instance. While the knife is the foreground and the chopping board is the background in the former, the chopping board becomes the foreground in the latter. The second example also shows the difficulty in determining which part should be considered as a foreground. The pan and the spoon were in confusion.

3.5.5 Cross-Dataset Transfer

To investigate how the features trained on the EK-Instance dataset generalize to novel examples, we conducted a zero-shot transfer from one dataset to another. We fixed the model and clustering algorithm to ResNet-34 trained with N-Pair loss and HAC. We used three additional datasets for the analysis.

Stanford Online Products [85]: This dataset is composed of 120,053 images from 22,634 online product categories collected from eBay. Following the original split, we used 11,318 categories for training and 11,316 categories for testing, while the hyperparameter was tuned by further partitioning the training set for validation.

INSTRE [100]: This dataset includes instance-level annotations of images taken from three domains: architectures, planar objects, and daily stereoscopic objects (*e.g.*, toys). The images are either taken from the Internet or manual image recording that are partitioned into three subsets (INSTRE-S1/S2/M). Notably, the manually collected images are intentionally captured in 25 backgrounds including both indoors and outdoors to obtain adequate distinctive backgrounds. It is composed of 23,070 images which contain one instance per image from 200 instances (INSTRE-S1/S2) and 5,473 images which contain two instances per image from 100 instances (INSTRE-M). We used the 11,011 manually collected images from 100 instances (INSTRE-S1) for our evaluation, which is further split into 50 categories for training and the other 50 categories for testing.

CORe50 [86]: This dataset includes image sequences of 50 hand-held objects belonging to 10 class-level categories, originally proposed for continual learning. We also evaluated this dataset because it has been collected in 11 distinct backgrounds with natural hand occlusion, which shares the challenges of our work. We sampled images in 1fps for each image sequence, which produces 550 image tracks in total. We used this dataset for all the 50 objects only for testing due to the limited number of instances. The hyperparameter was chosen by which achieves the best AMI across the trials. Since the provided bounding box was taken larger than the foreground object, 24 pixels around each of the 128-pixel squares were cut out to form input.

Results Table 3.3 shows the quantitive results. The left column shows source datasets and the top row shows target datasets. Numbers in the bold and blue text show the best and the second-best results across the source datasets, respectively. We first found that the model trained by Online Products and INSTRE did not generalize to EK-Instance. They showed degraded performance against the ImageNet baseline. Meanwhile, the model trained by EK-Instance showed better performance on Online Products and CORe50 compared to the ImageNet baselines while it showed slightly degraded performance on INSTRE. Online Products and INSTRE did not generalize to each other, both showing inferior performance against the ImageNet baselines.

We think that the difference came from the property of the data. First, the Online Products dataset was collected from an E-Commerce website. It contains multiple images of products per instance, but most images are accompanied with a very simple background (*e.g.*, transparent or monochrome) with limited variation. While it contains significant variations in the viewpoints, poses, and illumination, the images are not only identifiable from their foreground appearance but also from the background appearance, which shortcuts the true difficulty of identification by the foreground. The significantly degraded performance in CORe50 dataset supports this hypothesis since the dataset is designed to include shared backgrounds among instances. On the other hand, the INSTRE dataset was intentionally designed to capture diverse background images out of their natural context. Therefore, the benchmark required models to focus only on the foreground to distinguish objects. However, the number of objects is relatively small and they are selected from apparently distinctive objects such as toy characters. Although they are captured in various lighting conditions and poses (mostly 2D rotations), the viewpoints were strictly controlled to face the frontal part of each object, which is far from practical situations. As a result, the model trained with INSTRE showed the best performance on the CORe50 dataset but got the worst performance on the Online Products dataset which requires distinguishing fine-grained appearance changes. Although not perfect, the model trained with EK-Instance was able to identify instances with foreground and background variation because it contains the natural appearance changes in dynamic environments.

3.5.6 Visualization of Learned Representation

To better understand which area the model is focusing on, we visualized the instance-specific activation heatmap of the model using Grad-CAM [125]. Specifically, we applied Grad-CAM to the normalized softmax model for its simplicity. Although our main focus is to learn a generalizable representation against unknown instances, it gives us useful insights on which region is the model focusing on. Figure 3.8 shows the class-discriminative regions of training images derived from the last layer of the ResNet-34 backbone. The first and second columns are the input image and its support for the ground-truth category. The other columns are the top-5 "hard" classes which showed high instance classification probability and their corresponding supports.

The first finding is that the model learned to focus on the peripheral region of objects such as dishes and mugs (first to the third row, second column). We think the foreground clutters present in the EK-Instance dataset guided the model to focus more on the peripheral region. Similarly, the model avoided focusing on non-discriminative regions such as hands (fourth row). However, in some cases, we found sharp activation heatmaps not necessarily focusing on all the discriminative regions (fifth and sixth row). The model showed sensitivity against relatively smaller regions rather than focusing on all the discriminative areas. In addition, we found some classes that failed to recognize the foreground region, which leaves room for improvement (last row).

3.6 Discussion

Throughout the analyses, we have shown that the problem of object instance identification is yet solved in a dynamic and realistic scenario. We summarize the implications learned from the analyses for further development as follows:

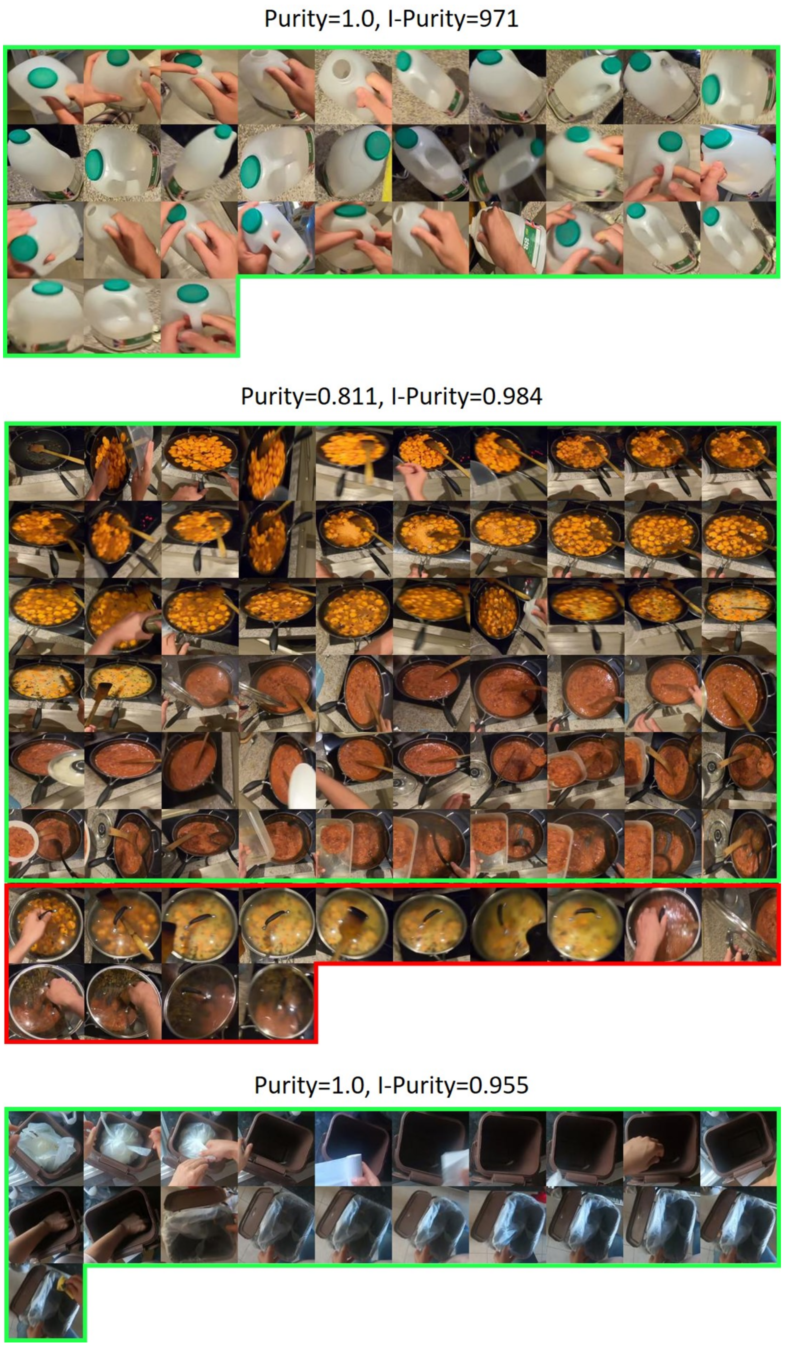
Appearance of objects dynamically changes across time: Prior works implicitly assumed that the object's appearance does not change upon reappearance. However, the EK-Instance dataset revealed that the appearance of objects will significantly change by the action of the user and the interaction between other objects. Towards a practical identification model, a model should learn not only invariance against viewpoint but also invariance against object state-changes. This aspect has been studied in a few works (*e.g.*, [126]) but overlooked in general. Although the latest metric learning-based models worked well in general, further analysis on those difficult cases should be conducted.

Both low-level and high-level features are important: In the EK-Instance dataset, we did not include class-level labels because it was difficult to provide a well-defined category set for it. However, the error analysis revealed that confusions among completely different semantic categories were present by giving too much focus on superficial resemblance. An explicit mechanism to combine both low-level and high-level features in a sophisticated way is demanded.

Foreground extraction matters: The main limitation of the EK-Instance dataset was that it lacks pixel-level annotation of the target object, which caused confusion in determining the foreground region of the target object. While it is unrealistic to provide fine-grained annotation against arbitrary categories, unsupervised techniques such as motion-based segmentation (*e.g.*, [127]) may mitigate the problem.

3.7 Conclusion

In this chapter, I have focused on the object side of hand-object interaction, and have studied the problem of category-agnostic object instance identification, which will be a critical component for long-term hand-object interaction understanding. By pointing out the limited data size and oversimplification in the existing datasets, I have newly collected the EK-Instance dataset, which contains more than 1,500 open-vocabulary object instance annotations from unscripted, real-world first-person videos. The collected dataset has shown larger intra-class variability by significant foreground and background clutters primarily caused by the user's activities. I have formulated the category-agnostic object identification problem as a clustering problem and provided strong baseline models based on metric learning. The experimental results have shown that the learned model exhibits better robustness against dynamic changes in appearance compared to the models trained by existing datasets. The analyses have revealed the existing biases in the trained model, and have discovered challenges for further development such as change robustness, feature fusion, and accurate object targeting. Future work includes (i) a new metric learning scheme that is more robust against appearance changes and (ii) scalable knowledge extraction from unlabeled videos.



Purity=1.0, I-Purity=971

Purity=0.811, I-Purity=0.984

Purity=1.0, I-Purity=0.955

Figure 3.6: Example of obtained clusters. Colored frames show images included in ground truth clusters. Purity and inverse purity [1] of each cluster are also shown.

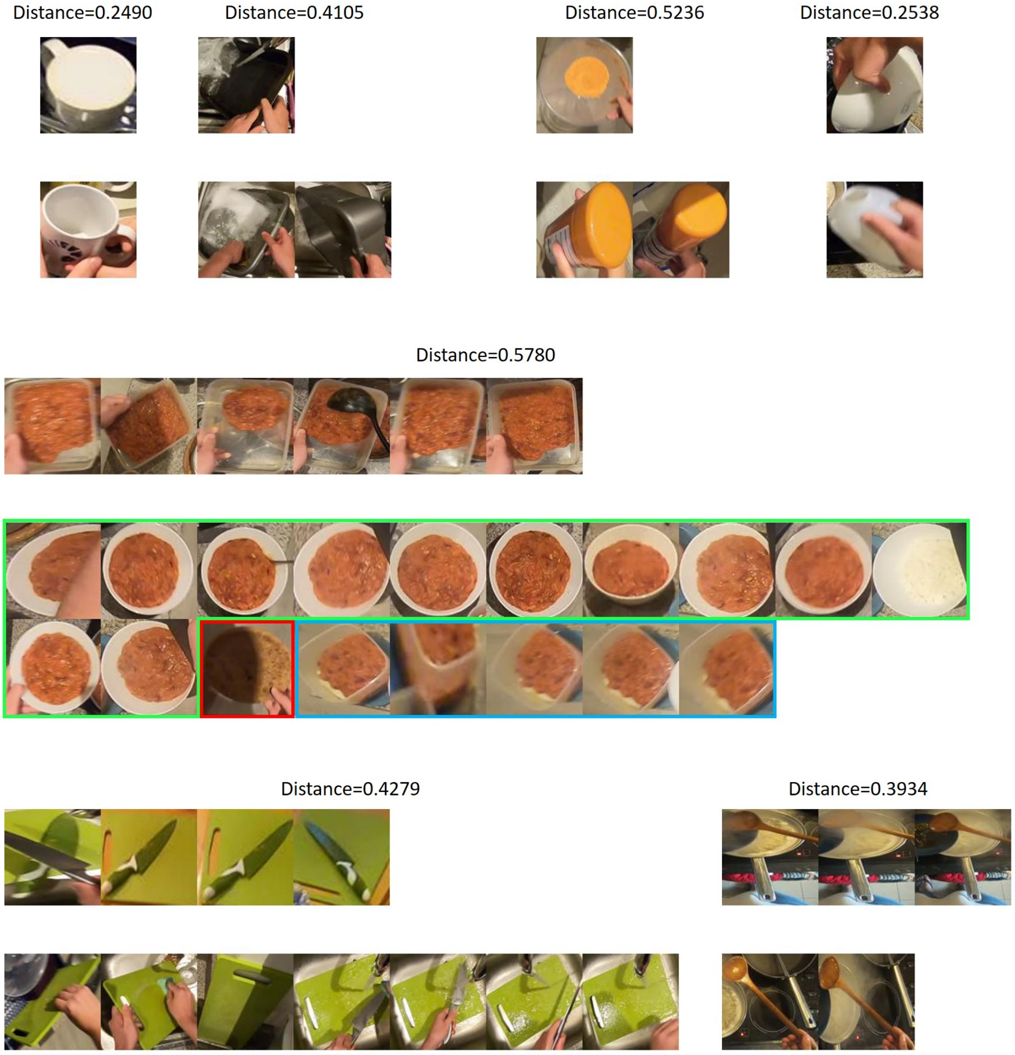


Figure 3.7: Failed HAC merging examples. First and second row of each item show tracks that are wrongly merged during HAC algorithm. Colored frames in middle denotes tracks from different instances. Averaged distance (1 - cosine similarity) between tracks are shown for each example (clusters are merged if distance is below 0.7).

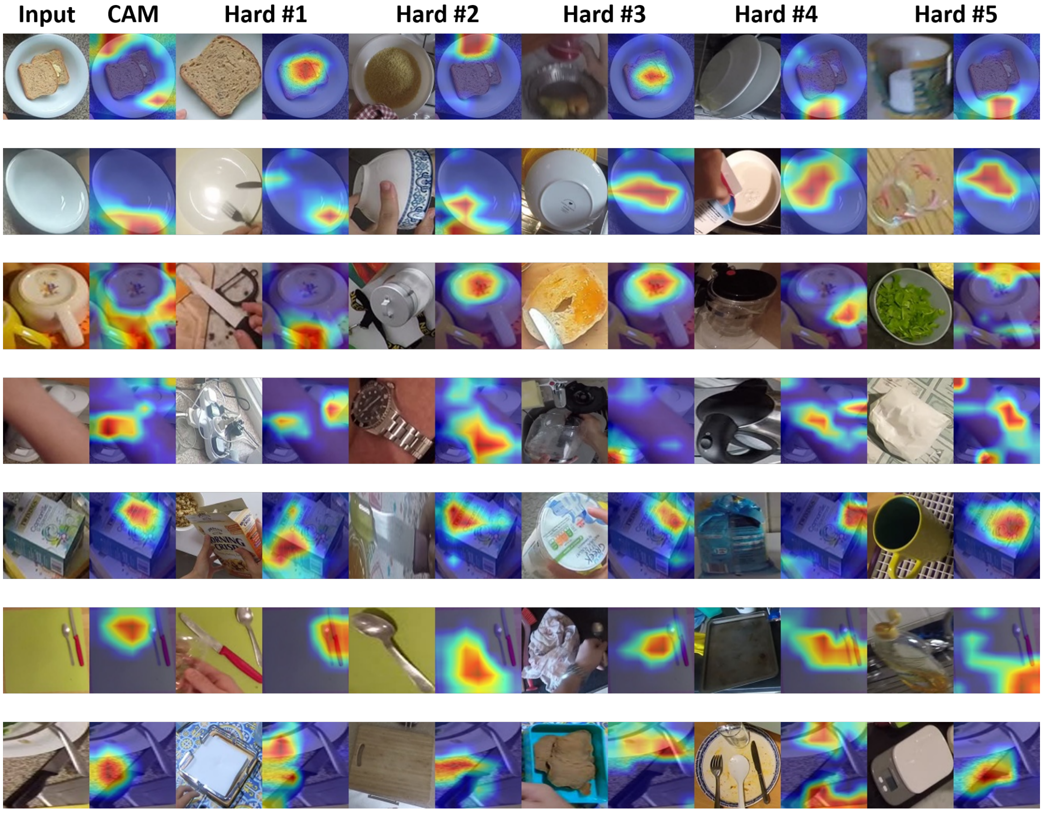


Figure 3.8: Grad-CAM visualization on training set. First and second column are input image and its support for ground-truth category. Other columns are top-5 “hard” classes which showed high instance classification probability and their corresponding supports, respectively.

Chapter 4

Assisting Users for Finding Lost Objects

4.1 Introduction

Looking for an object we do not remember leaving somewhere occurs frequently and is considered as a recurring problem regardless of age [128]. We lose objects under various reasons and situations [128–130]. One survey reports that people spend 2.5 days a year looking for misplaced objects [131]. As shown in the recent emergence of AirTag [132] for example, technological support to assist users in finding lost objects is demanded.

Ubiquitous computing tackles this problem by collecting and providing cues on where objects are located. Placing external sensors on the target object [133, 134], and detecting objects with visual sensors [135–137] are proposed as major solutions to keep track of object locations on behalf of users. Such prior systems are designed to track a small number of important objects and ask a user to register target objects in advance to track those objects. When looking for an object, the user searches a list of the registered objects (*e.g.*, a list of object names) to select which object to look for.

However, objects we lose are not necessarily registered. We often lose unique objects such as an important document received from the supervisor, a book borrowed from the library, or a thing bought a week ago which is left behind somewhere. Since such objects are not usually registered, the system

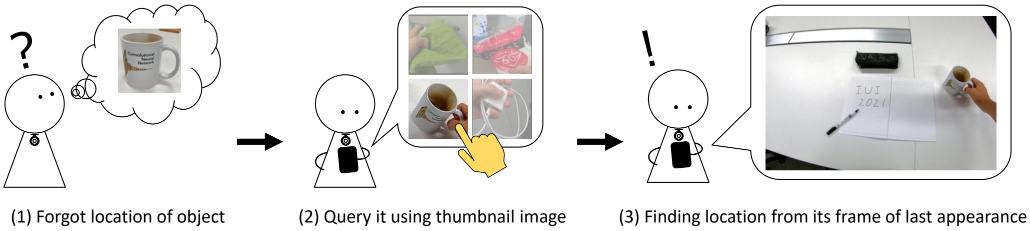

(1) Forgot location of object
(2) Query it using thumbnail image
(3) Finding location from its frame of last appearance

Figure 4.1: GO-Finder assists users in finding lost objects by showing last scene when the user handled it. User looks through list of object images to select object of interest.

cannot help users find them. To deal with such losses, we may think of automatically registering all the objects appearing around the user. However, this produces an enormous amount of candidates, which makes it impossible for the user to find an object within a reasonable amount of time. Moreover, assigning a unique name to each object will be unrealistic as the number of objects grows. To support finding arbitrary objects, we need not only to track potential objects to be lost but also to eliminate the burden of registration.

In this chapter, I introduce two key ideas to overcome the above issues to support users in finding lost objects. First, instead of tracking all the objects appearing around the user, we limit the search range to *objects handled by hands*. Since most portable objects we want to look for are handled with our hands, we can significantly reduce the number of candidate objects by limiting the scope to hand-held objects. A reduced number of candidates enables users to look for the target object in a realistic amount of time.

Another key idea is to *use the object image as a query* to select which objects to look for from the candidates. Instead of assigning unique names to objects, we display to users a list of object images to select which object to look for. Visual information of objects enables the user to identify the target object instantly without assigning a unique name to it.

Based on these ideas, I propose GO-Finder ("Generic Object Finder"), a registration-free wearable system for assisting users in finding arbitrary hand-held objects. GO-Finder only requires a video captured from the wearable camera, does not require any registration, and can handle arbitrary hand-held objects automatically. When finding objects by GO-Finder, users first skim through a list of object thumbnails, called *the hand-held object timeline*, to ask the system which objects to search for. Given the selected object, GO-Finder presents an image of the last scene when it appeared (Figure 4.1). This is achieved by a fully automatic process of *hand-held object discovery*, which detects and clusters hand-held objects.

To validate the effectiveness of GO-Finder, two studies are conducted on (i) user experience in a laboratory setting and (ii) usability study on interface in a longer and realistic scenario. In the first study, participants are to perform a object retrieval task in a laboratory setting, mimicking a situation of finding an object. A user study shows that users can successfully find objects by using the hand-held object timeline and reduce their mental load on performing the object-search task compared to the unaided condition. Participant feedback suggested that it is feasible to find arbitrary hand-held objects using the hand-held object timeline, which significantly broadens the coverage of objects to look for.

While the first study revealed GO-Finder's effectiveness on a relatively short sequence, the usability of GO-Finder on a much longer sequence with a large number of discovered objects was not yet investigated. In such a situation, the hand-held object timeline was expected to be suboptimal because the user must scroll through a very long list. To this end, we further propose two features that help users finding out the target object from the interface by providing the contextual information of it: (i) narrowing down by scene and time (ii) jumping to similar-looking objects. Users can efficiently browse the candidates by providing the context of the target object to the system.

As a second study, a usability study on the newly introduced features in situations where a large number of objects appear, along with algorithmic improvement, is conducted. The participants performed the task of finding out the target objects that appeared in a pre-recorded first-person video and how users benefit from context-based filtering was evaluated. User feedback confirmed the usefulness of context-based filtering against longer sequences while the simple hand-held object timeline was also shown to be more effective than expected.

4.2 Related Work

4.2.1 Computational Systems for Finding Lost Objects

Various types of sensors, such as wireless tags [136, 138–141], Bluetooth [142–144], stationary cameras [137, 145], and wearable cameras [135, 146, 147], have been studied for systems to assist users in finding lost objects. Active and passive radio-frequency identification (RFID) tags are frequently deployed by attaching them to target objects. While RFID tags are effective in indoor environments, they cannot locate an object when taken outside the search range. To expend the search range, a combination of Bluetooth and global navigation satellite system (GNSS) are adopted in some commercial products (*e.g.*, Tile [148] and AirTag [132]). Although these systems can provide the angle and distance from the tag, their guidance is less intuitive and attaching an external tag to each object will be a major bottleneck to track a large number of objects.

Alternatively, camera-based systems have the merit of not requiring external sensors attached to objects. Captured images themselves can be easily interpreted by the user just by showing the image [147] and visual information offer a great amount of information when remembering past events [149]. Butz *et al.* [145] used augmented reality (AR) markers to search for objects in an office environment. Xie *et al.* [137] proposed a dual-camera system for indoor object retrieval. Cook's collage [150] propose a system to monitor the progress of a cooking activity from a stationary camera. However, stationary cameras do not solve the problem of the search range and are weak against occlusions when objects are hidden by other entities. AR marker-based approach has the merit of target object easily recognizable by the camera but the registration cost is still high and limited to objects where AR marker is attachable.

Wearable camera-based systems mitigates these problems by capturing images from the user's viewpoint. Since the camera moves along with the user, the system captures a close-up of the surrounding environments and it can be carried, significantly expanding the search range. Similar to GO-Finder, Ueoka *et al.* [135] developed a wearable camera based object retrieval system based on object detection. The system consists of head-mounted RGB and infrared cameras for capturing pre-registered objects. It assists in object search by showing the last scene of the target object detected. The same strategy is adopted in this work. But unlike [135], our wearable-camera-based system, however, automatically groups all the hand-held objects appearing around the user, eliminating the registration operation.

Different from all the above works assuming a small number of items to be manually registered, we tackle the challenging problem of fully-automatic hand-held object tracking. We provide the users with how to select the objects of interest efficiently from a list of automatically tracked objects. Since GO-Finder automatically detects and groups all the hand-held objects appearing around the user, it can support the user even they lose unregistered objects.

4.2.2 Camera-based Systems for Mitigating Memory Problem

Camera-based systems are used for mitigating memory problems other than losing objects since visual information offers a large amount of information better than textual information [149]. Schiele et al. [151] proposed a wearable system to associate objects and videos taken from a body-worn camera to recall information of objects. Tran *et al.* [150] proposed a system to monitor the progress of a cooking activity. Hodges *et al.* [152] proposed a wearable camera based system called SenseCam, which takes wide-angle pictures periodically (*e.g.*, one shot every 30 s) to remind users of past events. Li *et al.* [153] proposed FMT, a wearable memory-assistance system to remember the state of objects (*e.g.*, the last time the plant was watered). While their hardware configuration is similar to ours in using neck-mounted wearable cameras, they aim to recall past interactions of a few numbers of daily-used objects, asking users to attach AR markers to each object. In contrast, GO-Finder aims to expand the range of objects which could be searched for by removing the registration operation.

4.2.3 Objects and Hands in First-Person Videos

GO-Finder executes hand-held-object detection and grouping to discover objects appearing in first-person videos. Discovering objects in first-person videos is a difficult problem since object categories appearing in daily life are massive, diverse, and individual-dependent. To this end, various methods have been proposed to discover objects in first-person videos [154–157]. Lee *et al.* [154] developed a model to discover important object regions using multiple first-person saliency cues. Lu *et al.* [158] proposed an object clustering-based method for personal-object discovery. Their system involves object-scene distribution based on the assumption that personal objects appear in different scenes while non-personal objects typically remain in similar scenes.

Since objects appearing in first-person videos are typically handled by hands, hand information is used to improve object detection. Higuchi et al. [159] propose to use hand appearance as one of the crucial cues to efficiently fast-forward first-person videos, which was further extended to extract important moments in a surgery [160]. Lee *et al.* [65, 161] proposed using hands as a guide to identify an object of interest from a photo taken by people with visual impairment. Shan *et al.* [22] collected a large-scale dataset of hand-object interaction along with annotated bounding boxes of hands and objects in contact with each other. Their proposed system can detect hands and objects in contact with each other from an image. Our aim is not only detecting hand-held objects but also to discover hand-held-object instances from first-person videos, which reduces the number of candidates to be registered.

4.2.4 Object Retrieval Behavior of Humans

When people try to find lost objects, instead of recalling the object itself they typically start by recalling contextual information around the object to look for. The field of Personal Information Management (PIM) [162] study how people behave on acquiring desired items or information. Elsweiler *et al.* [130] reports several recovery strategies when looking for misplaced objects revealed by a diary study. Participants reported a "spatial mental journey” recollecting through likely locations or events using visual and spatial contextual information. They tried to recollect the spatial and temporal contexts in which objects were used to make a guess on where to look for the misplaced objects. Kelly *et al.* [163] also reports that contextual information such as date, time, and location associated with the digital items is well remembered over six months. Indratmo *et al.* [164] suggests providing multiple visualizations as views so that the users can reach out to the target object from various perspectives, depending on their cognitive habits. These studies suggest that it is beneficial to access the context of the target object when finding lost objects.

Because our system aims to discover arbitrary hand-held objects to be searched, it produces a large number of objects to look for within the interface. As the number of discovered objects grows, it will be more difficult and inefficient to skim through a long list of candidate objects. To overcome this problem, we make use of the above practices by incorporating a function similar to these “filter-by-context” schemes into our interface.

4.3 System Design

4.3.1 System Overview

GO-Finder requires a wearable camera, processing server, and smartphone for browsing the location of objects the user is looking for (see Figure 4.2). The procedure is divided into observation and retrieval phases.

In the observation phase, a user wears a camera on their neck. The camera continuously stores the first-person images send to the processing server. The server processes the received images to detect and track hand-held objects. Finally, images are clustered by their appearance to discover groups of object instances.

In the retrieval phase, users use a smartphone-based interface (see Figure 4.3) to receive the processed results thorough a wireless connection. First, users select which object to look for through the hand-held object timeline (Figure 4.3 left). Then, they find the target object by viewing the pop-up screen showing the last appearance of it (Figure 4.3 right).

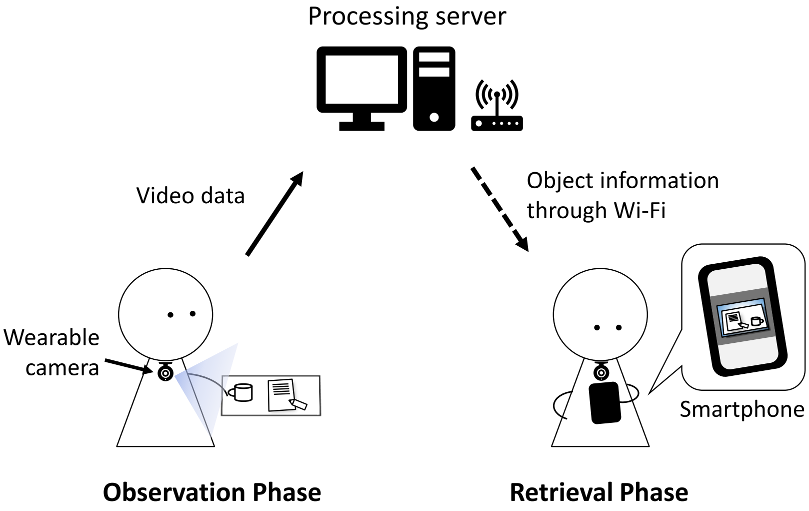


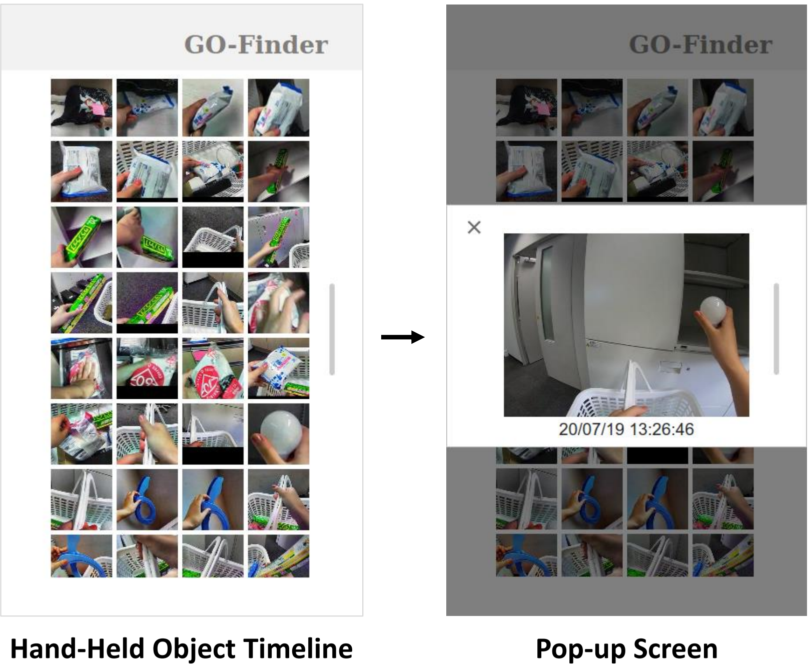
Figure 4.2: Users wears wearable camera on their neck. During observation phase, their first-person images are sent to processing server to discover hand-held objects. At retrieval, processed results are sent from server, and user retrieves last frame of objects through smartphone app.

4.3.2 Hand-Held Object Discovery

GO-Finder attempts to detect hand-held objects and discover groups of object instances from the first-person video. By discovering object instances, we can acquire the last appearance of the object, which is used to find the object. Figure 4.4 shows a rough sketch of how to acquire the last appearance of an object. An object detector detects hand-held objects from first-person video frames. From all the detected object images, we apply tracking and clustering (see Section 4.4 for details) to discover groups of cropped object images, clustered by instance. Since we are interested in finding the last location of the object, we only use the last thumbnail image and last frame for our user interface.

4.3.3 Hand-Held Object Timeline

GO-Finder automatically discovers hand-held object instances and registers them as candidates. In this case, searching for objects by their names be-



Hand-Held Object Timeline

Pop-up Screen

Figure 4.3: Interface of smartphone app used in first study. (Left) Hand-held-object timeline. (Right) Pop-up screen.

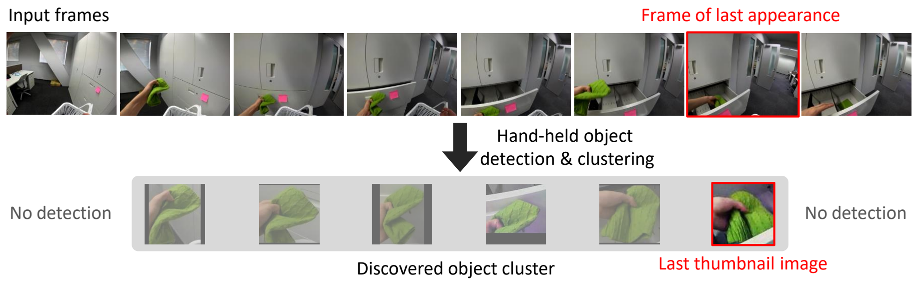


Figure 4.4: Given first-person video frames, system detects hand-held objects and groups them to discover cluster of cropped object images for each object. Since we are interested in providing last location where the object appeared, we use last thumbnail image and last frame in which the object appeared to help user find specific object.

comes unrealistic since it requires an association between the object name and its appearance. We propose *the hand-held object timeline*, which selects the target object by browsing the thumbnail images of the objects placed over a grid (see Figure 4.3 left). Thumbnail images of the objects are sorted by the last time they appeared in descending order. By skimming through the timeline, users select a thumbnail of the target object to retrieve its last appearance. We adopt the image timeline as a metaphor for a photo album, which is widely accepted in existing smartphone-based interfaces.

Note that the obtained object timeline can be used as a trigger to remind the user of the object location. The timeline acts as a concise history of what the user has handled in the past. Even before arriving at the target object, the user can be reminded of past actions by looking back at the timeline.

4.3.4 Pop-Up Screen

By clicking on a thumbnail of the object timeline, a pop-up screen will appear to show the appearance of the object and time (see Figure 4.3 right). Since the pop-up screen shows the critical moment of leaving an object, the user can instantly be reminded of the location of the object by looking at the surrounding environments. The "basic" version with minimum functionality that combines the hand-held object timeline and the pop-up screen was used

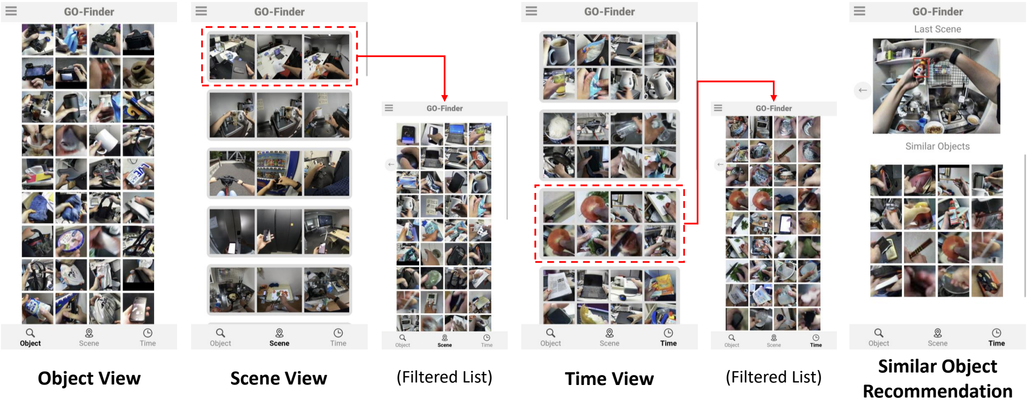


Figure 4.5: Extended interface with candidate object filtering and recommendation, used in second study.

in the first study.

4.3.5 Candidate Filtering by Context

While the hand-held object timeline is simple and intuitive, the browsing cost linearly increases to the number of objects. As the number of objects grows, scrolling through a very long list will be more difficult. Based on the typical object retrieval strategies reported in PIM, we explore the possibility of providing users options to filter the candidate objects from the contextual information (*e.g.*, time, location, and co-occuring objects) they remember.

Specifically, three additional features (scene view, time view, and similar object recommendation) are introduced as an extension to the basic version (see Figure 4.5), that are used in the second study. In this "extended" version, the user can select how to find the objects from three views. Furthermore, a recommendation function that allows the user to search the target object by its appearance is introduced.

Object View

This view is identical to the hand-held object timeline. All the thumbnail images of the appeared objects are sorted by the last time they appeared in descending order.

Scene View

In the scene view, discovered objects are grouped by scenes where the objects appeared. Scenes are automatically discovered by a way similar to hand-held object discovery. Several frames which represent the scene are displayed per scene, and by pressing one of them, a filtered list of objects which appeared in that scene will be displayed the same as the interface of the object view. If the user partially remembers the location the target object appeared, they can filter a large number of candidates by the selected scene. We note that if an object appears in a scene at least once, the object will be included in the scene. Therefore, even when the user does not know the last appeared location, they can find the object from the past locations the object appeared.

Time View

Similar to the scene view, discovered objects are divided by a set of fixed length time windows. Several object thumbnail images which represent the time window are displayed per window, and by pressing one of them, a filtered list of objects which appeared within the time window will be displayed the same as the interface of the object view. The time view is effective when the user remembers the time or the activity when they handled the object. The difference appears when the user arrives at the same scene at a different time. By using the time view, the user can narrow down the objects associated with that specific moment.

Similar Object Recommendation

This feature is implemented upon the pop-up screen. A list of object thumbnails that show high visual similarity to the selected object is displayed bottom of the screen. The user can jump to the pop-up screen of similar-looking objects, which allows the user to navigate to the desired object even they cannot find it from the hand-held object timeline. This process could be repeatedly applied thus the user can jump to other objects more than once. Similarity scores are computed by the same model used for the hand-held object discovery.

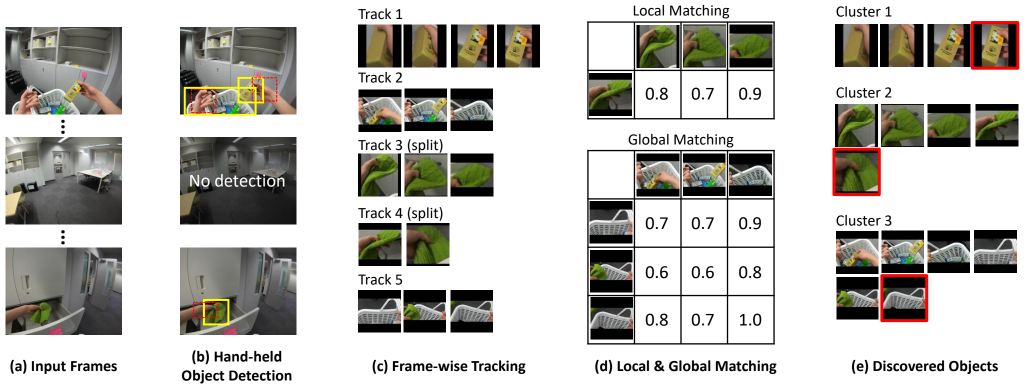


Figure 4.6: Overview of hand-held-object-discovery algorithm. (a) Input frames. (b) Example of hand-held object detection. Yellow and red boxes denote detected objects and hands, respectively. (c) Tracked detections. Typically, they are segmented due to tracking failure or re-appearance. (d) Local matching between latest detection and existing cluster (top). Global matching between two existing clusters (bottom). (e) Segments are clustered by instance. Last appeared scenes (images with red frame) will be displayed in user interface.

4.4 Algorithm and Implementation

In this section, we introduce the details on the hand-held object discovery algorithm used in GO-Finder.

4.4.1 Hand-Held Object Detection

We use the state-of-the-art algorithm on hand-held object detection [22] trained on a large-scale image dataset of hand-object interaction collected from first-person video datasets [20, 21, 28]. Given a video frame, it produces bounding boxes of the hand, contact state (self-contact, other people, portable object, and static object), and its manipulating objects (see Figure 4.6 (a)). Since various types of objects under a hand contact are annotated, the model can detect arbitrary types of objects in contact, while rejecting other objects not handled by hands. Therefore, we can significantly reduce the number of candidates to be searched compared to detecting all the objects that appear in a scene. Since we are interested only in portable objects, we extract detections that are predicted as a portable object in the contact state prediction. Furthermore, detections that occupy more than half the side length of the frame are considered noise and are excluded from prediction.

4.4.2 Object Instance Discovery

Using the detected bounding boxes, we cluster them into a set of instances based on their appearance features. Every detection should be assigned to a single cluster, and re-appearing objects should be merged into existing clusters. To this end, we adopt a combination of local and global matching, which consists of three stages.

Stage 1: Frame-wise Tracking

We first apply a visual tracker to the detected hand-held objects. If the tracker successfully associates between consecutive detections, we assign the detection to the same cluster as the previous one (Figure 4.6 (c)). Since first-person videos include large camera motion, we use an appearance-based tracker [165], which performs similarity matching. The cost assignment matrix is calculated by the intersection-of-union between all the tracker's predictions and actual detections. Optimal assignment is achieved by using the Hungarian algorithm [123].

Stage 2: Local Feature Matching

When the tracking fails, we apply local matching between the latest detection and existing clusters based on the object's appearance. We use pre-trained convolutional neural network (CNN) features to find similar objects in the existing clusters. For every detection, a 2048-dimensional feature vector is first extracted from the layer before the final layer of ImageNet-pretrained ResNet-50 [116]. We then calculate the cosine similarity between the new detection and all the detections in the cluster (see Figure 4.6 (d), top). Next, for each cluster, if the maximum and median of the similarity scores are above certain thresholds, the new detection is merged with that cluster. We check the median score to avoid false associations. If none of the clusters meets the condition, then a new cluster is created.

Stage 3: Global Cluster-wise Merging

Since the previous stage matches against a single detection, it tends to form a new cluster if the viewpoint or boundary of the latest detection fluctuates even it should be merged. To deal with such incorrectly segmented clusters, we try to merge clusters by global cluster-wise merging. Given a pair of clusters, we calculate sample-wise cosine similarity between clusters, forming a similarity matrix (see Figure 4.6 (d), bottom). Note that the scores calculated at stage 2 can be reused in this stage. The two clusters are merged if the maximum and median of all elements of the similarity matrix were above certain thresholds. Concretely, the maximum and median thresholds are set to 0.8 and 0.7, respectively.

However, this merging process is time-consuming, and should not be repeated every time. To reduce the number of trials, we re-try merging only if the number of similarity matrix elements is more than two times that of the last trial.

Determining Similarity Thresholds

Changing the hyperparameters (maximum and median similarity threshold) may affect user experience. Stricter thresholds produce oversegmented and increased number of clusters while achieving higher recall on discovered target objects. This makes it more difficult for the user to select the object of interest from the candidates. In contrast, looser thresholds result in a smaller number of clusters with the risk of missing objects due to wrong associations. A reduced number of clusters may make it easier for the user to select the target object, but it may be impossible to find it if it is incorrectly merged with other objects. While we empirically selected these parameters during the study, we further introduce additional heuristics to explicitly suppress false associations.

Constrained Clustering using First-Person Cues

During similarity calculation, the hand-held object discovery algorithm sometimes shows a high similarity to a different object due to the appearance of the hand and similar textures, producing false associations. Therefore, we introduce several heuristics to suppress such false associations. If a detected bounding box or a pair of them satisfies the following conditions, the similarity of that pair is set to zero.

- **Aspect ratio between two boxes:** if the ratio of the two bounding box aspect ratios is larger than 1.5
- **Ratio of skin color:** if the ratio of the skin-colored region (calculated using color histogram) is larger than 0.3
- **Area ratio of the object to the corresponding hand:** if the ratio of the two area ratios (area of the object bounding box to that of the hand bounding box) is larger than 1.5

The above detection-and-discovery algorithm was used in the first study.

4.4.3 Dealing with Real-World Environments

While the pre-trained CNN model introduced in 4.4.2 works reasonably well in discovering object instances, its performance is not enough when many similar-looking objects appear in a cluttered environment. To further reduce under/over-segmentation in real-world environments, a metric learning model is introduced along with modifications on the previous algorithm.

Metric Learning by Tracking

The training data is automatically generated by hand-held object detection and tracking. Given a video clip, we first use the hand-object detection model [22] to detect hand-held objects in each frame. Next, we associate adjacent detections using a Kalman Filter-based tracker [52]. We further extrapolate object tracks one second before and after using another visual tracker [53].

Since the generated hand-held object tracks include pose change and occlusion during the interaction, we can learn an embedding robust against pose change and occlusion by using them as a training data. Specifically, we employ the well-known triplet loss [166] which is popularly used in re-identification tasks. Given a projection model f_θ parametrized by θ, a distance metric function D, an anchor point $\mathbf{x}_a$, a positive point $\mathbf{x}_p$ sampled from the same track, and a negative point $\mathbf{x}_n$ sampled from other tracks, the objective is to make the distance between projections of $\mathbf{x}_a$ and $\mathbf{x}_p$ closer than the distance between projections of $\mathbf{x}_a$ and $\mathbf{x}_n$ by at least a pre-defined margin m.

$$L_{\text{triplet}}(\theta) = [m + D(f_\theta(\mathbf{x}_a), f_\theta(\mathbf{x}_p)) - D(f_\theta(\mathbf{x}_a), f_\theta(\mathbf{x}_n))]_+.$$

We select positive points from the same track the anchor point belongs to while selecting negative points from the other tracks.

We generated the training data using the EPIC-KITCHENS dataset [23, 28] which contains cooking activities from various environments. Because diverse objects appear in the kitchen environment, we found the network learns generalizable features for instance re-identification. Because videos are divided by participants, we collected negative points from videos of a different participants.

In this version, we use an ImageNet [167] pretrained BN-Inception [168] CNN followed by a linear projection layer (128 dims) and an L2 normalization layer as a projection model. BatchNorm parameters are fixed during training to prevent overfitting. The Euclidean distance is used as a distance metric function, which is equivalent to cosine similarity for L2-normalized embedding.

Rejection by Semantic Information

While the metric learning-based model generally produces better similarity measures, it suffers from wrongly associating objects of completely different categories by emphasizing color and texture information. To eliminate such false associations, we introduce another CNN model which performs matching by semantics. Specifically, we calculate the class probability of the detected object image using an ImageNet [167] pretrained BN-Inception [168]

CNN network. Kullback–Leibler divergence is used to calculate the semantic similarity between detections. We expect that the same instance will produce a similar class probability. At local and global matching, we reject the merging regardless of the appearance similarity if the mean Kullback–Leibler divergence between clusters is above a certain threshold. In this study, we set the threshold to 0.004.

Cluster Size Limitation

In real-world scenarios, few objects appear frequently and their object cluster size grows quickly. This results in increased feature matching cost and large memory consumption, making real-time clustering impossible. We found that setting a maximum size on each cluster significantly reduces the computational cost without damaging the performance. When we found a cluster its size exceeding a certain threshold, we uniformly re-sample the feature vectors sorted by the corresponding detection's time. In this study, we set the maximum cluster size to 500.

Improved Thresholding

The original algorithm tends to oversegment temporally adjacent detections due to pose and background change. We add a heuristic of increasing the similarity of two samples with a predetermined offset when the samples are detected within a certain temporal range (*e.g.*, 10 sec).

4.4.4 Scene Discovery

To provide users the ability to filter candidate objects by scene, we also automatically extract a "scene" defined as a set of frames that their appearance is similar. We follow a similar strategy to the hand-held object discovery algorithm. We aim to extract scenes based on the workspace in which each action was performed (*e.g.*, kitchen sink, desk, and bookshelf) because appearing objects may completely change across workspaces even their spatial distance is small. Based on these demands, we introduce a local descriptor-based scene representation and clustering for scene discovery.

Scene Descriptor

We use Vector of Locally Aggregated Descriptors (VLAD) [169] to represent a scene. First, keypoints are extracted and described by Scale-Invariant Feature Transform (SIFT) descriptor [170]. Each descriptor is assigned to a pre-calculated cluster and its residuals between descriptors and cluster centers are accumulated and summed to form a single VLAD descriptor. We follow [169] to form a normalized VLAD descriptor of a vocabulary size of 128, which was further projected down to 4096 dims via principal component analysis. We extracted the SIFT descriptors from a 640×480 image, setting a very small Difference-of-Gaussian (DoG) threshold to extract many descriptors from the entire image.

To adapt to first-person videos, we made two modifications in the local descriptor calculation. First, local descriptors within the lower 15% of each image are omitted because the lower region often contains the lower body of the user and the floor region, which do not characterize the scene well. The local descriptors within the hand region are also removed due to the same reason. We extract a binary hand region mask using a hand segmentation model trained on a first-person video dataset [20] and removed the local descriptor within and around the predicted hand regions.

Scene Clustering

Then we group the calculated features by their similarity. An ideal scene should preserve the temporal locality while separating the cluster when a large appearance change occurs. We apply Spectral Clustering [111] by constructing an affinity matrix based on the VLAD descriptors. We calculate the L2 distance between the descriptors to determine the similarity. Different thresholds were used to connect temporally adjacent nodes and to connect the other nodes.

4.4.5 Implementation Details

We sampled video frames at 10 fps, and further resized them into VGA resolution before processing. While a smaller frame rate is enough to capture

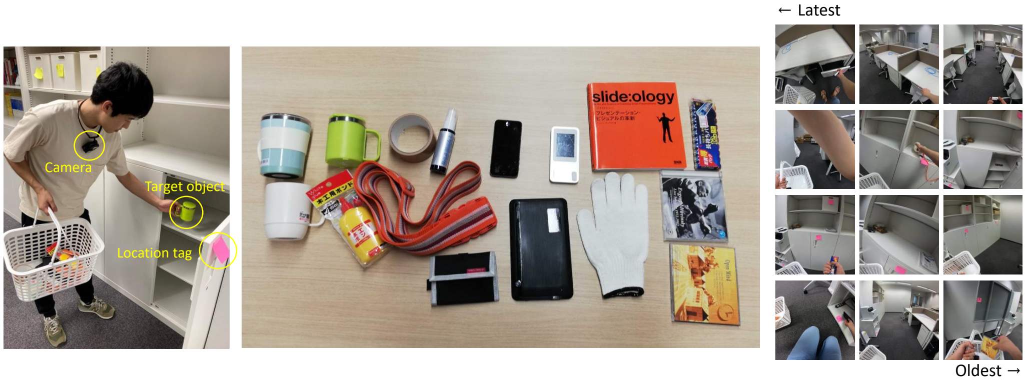


Figure 4.7: (Left) Object arrangement: participants hide objects at specified locations while wearing camera around their neck. (Middle) Objects used in study. (Right) Example of timeline of frame-based system.

the timing of leaving an object, we find that a higher frame rate is better to track objects stably.

4.5 Evaluation Studies

To validate the effectiveness of GO-Finder, we conducted two studies on (i) user experience evaluation in a laboratory setting and (ii) usability study on the interface in a longer and realistic scenario.

4.5.1 Study I: Finding Objects with GO-Finder

As a first study, we conducted an in-lab experiment to determine (i) whether GO-Finder can correctly discover hand-held objects from the video and (ii) whether users can use the system to find target objects. We hypothesized that by using GO-Finder, users can find objects correctly and quickly with less mental load.

Procedure

We recruited 12 volunteers[1] (10 males and 2 females) with ages ranging from 18 to 28. They were all familiar with using smartphones. The experiment was conducted in a room in our lab. The task was a *hide-and-seek* task performed by the participants. The procedure was divided into three phases: arrangement phase, forgettng phase, and retrieval phase. First, participants filled out a pre-study questionnaire on their past experience of looking for lost objects. After an introduction to the task, each participant was asked to hide a set of objects inside a room (arrangement phase), conduct a surrogate task to forget the locations of the objects (forgetting phase), and later asked to correctly retrieve a subset of them (retrieval phase). The trial was repeated three times, changing the experimental conditions. Conditions were randomly shuffled to eliminate order effects. After all the trials, participants filled out a post-study questionnaire on the usability of the interface. Finally, we conducted a semi-structured interview to find further insights.

Arrangement phase First, the participant went to the room and asked to hide a set of objects prepared by the experimenter. The locations to hide the objects were specified with pink tags and the participants were informed about them in advance (see Figure 4.7 left). The participants carried a basket along with the objects. During the experiment, participants wore a GoPro HERO 7 camera (150° diagonal field-of-view) to record first-person videos.

Forgetting phase The participants moved to another room and took a 15-minute interval to forget the arrangements. During the interval, the participant was asked to solve as many of a series of simple calculation problems as possible.

Retrieval phase The participants came back to the room and were asked to bring back a subset of the hidden objects. The list of objects to bring back was shown in a photo. In addition to the neck-mounted camera, the

[1] One participant (P05) was excluded from the analysis due to a misunderstanding of instruction.

participants wore smartphones around their necks to use the system. Under each condition, participants were given instructions on how to use the system and become familiarized with the interface by browsing the result of a sample video carrying a few objects. They were not forced to use the system; they used the system only when they needed to use it. All the experiment was completed on a Google Pixel 4 smartphone with a 5.7-inch, 1080×2280 pixel display.

Experimental Conditions

We compared three conditions:

- **No aid**: The participant search for objects themselves without any assistance.
- **Frame-based aid**: The participant is shown a timeline of images extracted every 5 sec.
- **Object-based aid (GO-Finder)**: Our proposed system with hand-held object timeline and pop-up screen.

The frame-based aid condition resembled automatic image capture devices such as SenseCam [152]. We hypothesized that past images would help the participants remember their arrangement of objects. Regarding the duration of the task, we showed images taken every 5 sec (see Figure 4.7 right), which is denser than typical devices (*e.g.*, 30 sec).

We used a laptop PC to run the object discovery algorithm. The connection between the laptop PC and smartphone was established via Wi-Fi as shown in Figure 4.2. At every trial, participants hid 16 objects in a choice of 20 locations and asked to retrieve 6 objects from them. We used different object sets for each trial, resulting in 48 objects in total (see Figure 4.7 right). The objects differed in color and shape, and sometimes included multiple instances of the same category.

System Evaluation Measures

Detection recall rate To measure how well the hand-held object discovery algorithm can detect target objects, we counted the number of objects that are successfully detected at their last appeared timings. The detection recall rate is defined as a ratio of the number of detected objects to the total number of target objects. It was manually calculated this metric by looking at the raw detection results. We counted as a success when there exist a bounding box well covering the object at least one frame. In combination with the localization rate (defined later), this metric can be used to measure whether the error exists in the detection phase or the clustering phase.

Localization rate To measure how well the hand-held object discovery algorithm can discover target objects, we counted the number of objects in which their locations are identifiable by a third person, who did not have any memory of arrangement; only using our system. The localization rate is defined as a ratio of the number of identifiable objects to the total number of target objects. It was manually calculated this metric by using the system. We counted as a success only if a close-up of an object is visible in the thumbnail of the timeline and the object location could be correctly determined from the pop-up screen without difficulty. This metric acts as an expected recall of the system.

Number of clusters We also measured the number of clusters formed with the hand-held object discovery algorithm and analyzed the contents of the timeline. We ran the algorithm for all 36 trials (12 participants × 3 conditions).

Objective Evaluation Measures

Correctness of retrieval We calculated the mean precision of each trial. We counted as correct when the user found an object listed on the target list and incorrect when the user opened a location with the incorrect or no objects. We compared three combinations of two of the conditions by using the paired t-test on the difference of mean scores.

Task completion time We expected shorter task completion time by using the system. We compared three combinations of two conditions by using the Wilcoxon signed-rank test on the difference in mean task completion times.

System usage time Since GO-Finder can search for objects directly through the hand-held object timeline, we expected to have a shorter usage time using GO-Finder to the frame-based aid condition. We measured the number of times participants used the system[2] and usage time per trial from the recorded videos.

Subjective Evaluation Measures

Questionnaire After all the trials, participants answered questions on each condition. First, participants were given the question, *"How do you rate the difficulty of completing the task?"* on a seven-point scale (easy = 1, difficult = 7). We used the Wilcoxon signed-rank test in the difference of means. Regarding the features of the interface, we asked whether they agreed to the following questions on a five-point scale: Q1) The timeline is easy to view. Q2) The timeline is intuitive to use. Q3) The timeline helped me look for objects. Q4) The pop-up screen is easy to view. Q5) The pop-up screen helped me look for objects. Q6) The timeline (under each condition) gave me a clue on the location of the target object. Q7) I could be reminded of the locations of the objects by using the system (under each condition). We also asked to answer the System usability scale (SUS) test [171]. Finally, we asked the question, "How comfortable was the neck-mounted camera?" on a seven-point scale (unpleasant = 1, comfortable = 7).

Observation and interview We observed how the participants searched for objects. During the interviews, we asked what they thought during the retrieval task. To collect insights on using this system in daily life, we also asked *"What do you recommend to improve the interface?"*, and *"How do you feel about wearing a camera in private/public places?"*.

[2] We counted as one time when the user attempted to search a location after using the system.

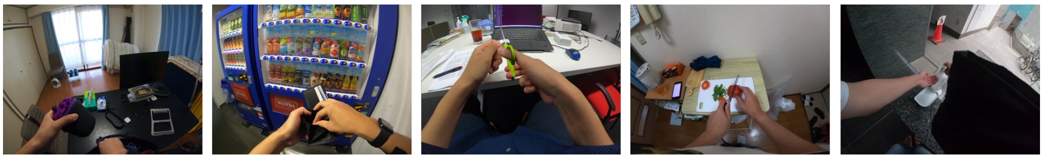


Figure 4.8: Example frames of video used in second study.

4.5.2 Study II: Usability Evaluation on Real-World Sequence

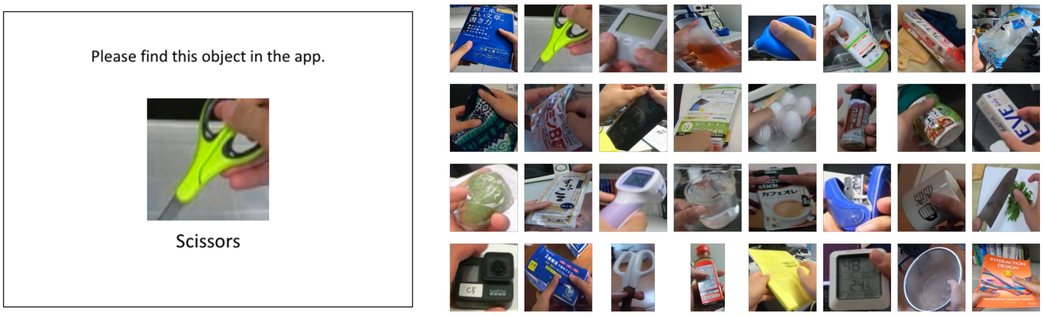
As a second study, we conducted a usability study on the extended interface in better realistic situations where a larger number of hand-held objects appear in a much longer sequence. As explained in 4.3.5, users can select which feature to use by their preference and can efficiently browse the candidates by providing the context of the target object.

In this study, we focused on collecting implications on the usefulness of each feature and the system's usability as a whole on selecting the desired items from a longer list. Because it is difficult to control participant's behavior for a long time, the participants are asked to watch a pre-recorded first-person video instead of performing an object arrangement task and then find out the target objects that appeared in the video using the system. We hypothesize that

- Users benefit from filtering the object candidates by the place or the specific moment they remember used the objects.
- Users benefit from the proactive recommendation of similar-looking items so that they can reach out to the desired items without scrolling the timeline.

Data and Implementation

We recorded a first-person video consisted of multiple scenes using a neck-mounted GoPro HERO 7 camera (see Figure 4.8). The video includes natural daily activities such as cooking a meal, making a drink, arranging objects, etc. The total length of the video was around 75 minutes (45,000 frames)


Please find this object in the app.

Scissors

Figure 4.9: (Left) Example of query. (Right) All used objects in second study.

and 145 unique hand-held objects appeared in total, significantly larger in terms of both video length and the number of objects.

We discovered objects using the improved algorithm and found 406 object clusters, about four times larger than that of the previous study. For the scene view, we calculated the dense VLAD descriptor every two seconds and obtained 30 clusters. Scenes such as in front of a sink, a desk, a table, and a vending machine were extracted. For the time view, we split the video by ten minutes, producing eight windows in total.

Procedure

We recruited 6 volunteers (4 males and 2 females) with ages ranging from 21 to 27. They were all familiar with using smartphones. First, the participant watched the recorded first-person video. They watched the video only once and re-watching was not allowed. After a ten-minute interval, the participant was asked to find the items that appeared in the video from the interface. Figure 4.9 left shows an example of queries shown to the participant. At each trial, the name and the appearance of an object are displayed as a query. The participant was instructed how to use each feature (*e.g.*, the scene view can filter the object candidates by which scene they appeared) on a short sample video recording and took a practice session on the sample data to get used to the interface.

We selected 32 items from the video as queries and the participant was asked to find them from the interface (see Figure 4.9 right for all the queries used in the study). The participants either used the “full” interface with all the features and the “baseline” interface identical to the basic version used in the first study. The order of the queries is fixed and the participants are grouped into two groups. The first group was asked to use the full interface on the odd numbered questions and to use the object-view only interface on the even numbered questions, and vice versa for the second group. After the task, the participants answered a questionnaire followed by a semi-structured interview on the usability of both interfaces.

Evaluation Metrics

Retrieval time For each trial, we measured how long does it take to reach the pop-up screen of the target object. Because of the oversegmentation, some objects are divided into multiple clusters. In such cases, a trial was considered successful if the user reached one of the correct clusters. We expected a shorter retrieval time using the full interface.

Questionnaire After all the trials, participants answered questions on how they used the interface under each condition and the usability of the features (object/scene/time view and recommendation). For each feature, participants were asked whether they agreed to the following questions on a five-point scale (Q8-19): (i) (Feature) is easy to view (ii) (feature) is intuitive to use (iii) (feature) helped me looking for objects.

Observation and interview We observed how the participants used the interface. In the interview, we asked what they thought during the task.

4.6 Results

4.6.1 Results of Study I

System Evaluation

Detection recall rate Table 4.1 shows the detection recall rate of the hand-held object detection algorithm. The average score was 94.3, 94.8, and

Table 4.1: Detection recall rate of each object set (%).

	Mean ± SD	Min	Max
Set 1	94.3 ± 5.6	81.2	100
Set 2	94.8 ± 7.5	75.0	100
Set 3	95.9 ± 4.9	87.5	100
All sets	95.0 ± 5.9	75.0	100

Table 4.2: Localization rate of each object set (%).

	Mean ± SD	Min	Max
Set 1	84.9 ± 7.8	68.8	93.8
Set 2	83.3 ± 9.7	68.8	100
Set 3	88.5 ± 10.9	62.5	100
All sets	85.6 ± 9.6	62.5	100

95.9% for each object set, and the overall average was 95.0%. These results indicate that around one object was missed per session and failed to produce the correct location. We observed small variance between the object sets on detection.

Figure 4.10 shows example frames when the target objects are missed. The detection algorithm failed to detect the target objects shown in the magenta boxes. We found few notable patterns in missed detections. Although the hand-held object detection algorithm stably worked when objects are grabbed by hand, there were few errors when the hand movement is rapid or the hand exhibit irregular poses (see Figure 4.10 left). Also, the algorithm failed to detect objects when they were not handled by hand (see Figure 4.10 middle and right). In these cases, the participants hid the object by not placing the object but by moving the basket. The detection algorithm failed because it required the object to be touched by hand.

Localization rate Table 4.2 shows the localization rate of the hand-held object discovery algorithm. The average score was 84.9, 83.3, and 88.5% for

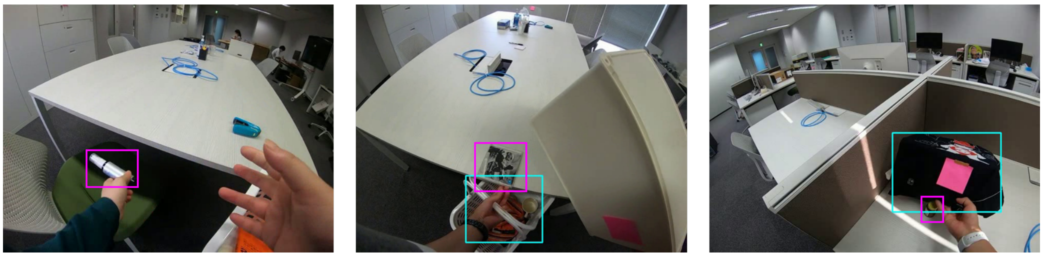


Figure 4.10: Examples of failed object detection. Cyan and magenta bounding boxes denotes detected object bounding boxes and missed object bounding boxes, respectively.

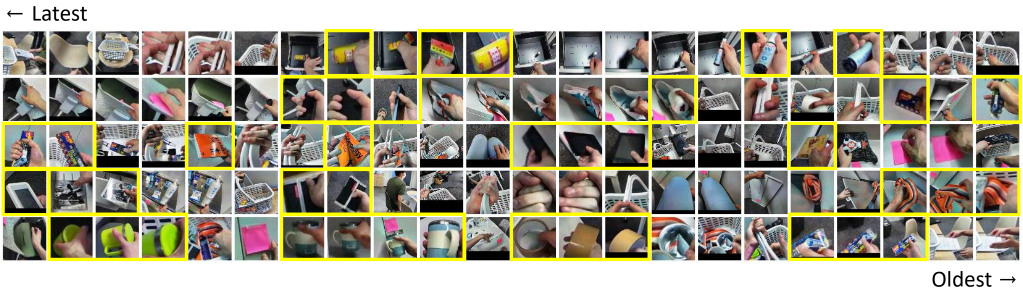


Figure 4.11: Example results of hand-held object timeline (localization rate=0.9375, #clusters=110). Yellow boxes denote clusters that contain target objects. Some clusters were over-segmented into few clusters per object.

each object set, and the overall average was 85.6%. These results indicate that GO-Finder can correctly display 13.6/16 objects per session on average. The minimum rate across the participants were 62.5%.

We found differences in performance among objects. While several objects were discovered in all 12 trials (green cup, wood glue, electric bulb, futon pincher, green cloth, and teddy bear), some objects were difficult to discover (waiter's corkscrew: 16.7%, medicine bottle: 58.3%, black wallet, and spray bottle: 66.7%). We found that small and black objects were difficult to correctly discover due to occlusion and texture-less regions.

We can analyze where errors occurred by looking at the difference between the detection recall rate and the localization rate. The results show that around 1/3 and 2/3 of the objects are missed at the detection and clustering phase, respectively.

Table 4.3: Retrieval performance (precision).

	Mean	95% CI
No aid	0.728	0.553–0.902
Frame-based aid	0.736	0.612–0.860
Object-based aid	0.922	0.838–1.006

Number of clusters The number of clusters (objects) that appeared in the hand-held object timeline was 108.6 ($SD = 24.0$) on average. Although it is not trivial to count the number of valid objects which should be discovered, the estimated number of valid objects (including furniture, drawers, and baskets) was expected to be about 20 to 30, including the 16 target objects. Thus, we can conclude that the algorithm over-segments an object into four to five clusters on average.

Qualitative analysis Figure 4.11 shows an example of the obtained hand-held object timeline from one trial. We annotated the thumbnail images that contain close-ups of the target objects in green boxes. Forty out of 110 clusters contained 15 of the target objects, one missed. In addition of the target objects, GO-Finder discovered various valid objects and false positives. Examples of valid objects were chairs, baskets, and drawers while untouched furniture, participant's body, and other people were discovered as false positives. While most objects were easily identifiable from the thumbnail images, some thumbnails were difficult to identify due to occlusion, shadow, and irregular views (*e.g.*, the green cup in Figure 4.11 left-bottom).

Object Retrieval Performance

Tables 4.3 and 4.4 show the results of the object retrieval task under each condition. We report the average precision and its 95% confidence interval (CI) under each condition. As expected, GO-Finder showed better precision with less variance than the other two conditions. The paired t-test revealed significance only between the frame-based aid and object-based aid conditions ($p = 0.918$, $p = 0.069$, and $p = 0.039$, respectively). However,

Table 4.4: Results of paired t-tests on difference in mean precisions.

			95% CI		Effect size
	t	p	LB	UB	d
No aid/frame-based aid	-0.104	0.918	-0.193	0.175	0.04
No aid/object based-aid	-2.012	0.069	-0.407	0.018	**0.95**
Frame-based aid/object-based aid	-2.339	**0.039**	-0.360	-0.011	**1.17**

Table 4.5: Task completion time (sec).

	Mean	95% CI
No aid	216	74–358
Frame-based aid	238	163–313
Object-based aid	178	126–230

both no aid/object-based aid and frame-based aid/object-based aid conditions showed large effect sizes ($d = 0.95$ and $d = 1.17$, respectively), indicating a positive effect by using the proposed system. In contrast, we did not observe a marked difference between no aid and frame-based aid conditions ($d = 0.04$).

Table 4.5 and 4.6 shows the result of the task completion time in each condition. We did not observe improvement in task completion time by using GO-Finder. The paired t-test did not show any significant difference ($p = 0.379$, $p = 0.309$, and $p = 0.077$). The average time and 95% confidence interval of the arrangement phase was 223 ± 16 sec.

Usage time During the 12 sessions under the frame-based aid and object-based aid conditions, participants used the interface 32 and 35 times, respectively. The mean (median) usage times were 28.1 sec (23.0 sec) and 16.1 sec (12.5 sec), respectively. The paired t-test revealed a significant difference with medium effect size in the mean times ($p = 0.005, d = 0.71$). This suggests that participants were able to browse the timeline more efficiently under the object-based aid condition than under the frame-based aid condition.

Table 4.6: Results of paired t-tests on difference in mean task completion times.

			95% CI		Effect size
	t	p	LB	UB	d
No aid/frame-based aid	-0.316	0.379	-173	130	0.12
No aid/object-based aid	0.513	0.309	-128	206	0.23
Frame-based aid/object-based aid	1.530	0.077	-27	148	**0.60**

Questionnaire

Ease of task Figure 4.12 and Table 4.7 show the results on ease of the task. Surprisingly, the participants evaluated the frame-based aid condition the most difficult. They evaluated the object-based aid condition the easiest among the three conditions. Based on the Wilcoxon signed-rank test, we found a significant difference in the mean scores between the frame-based aid and object-based aid conditions ($p = 0.063$, $p = 0.133$, and $p = 0.043$). However, we observed medium effect size in all the combinations ($r = 0.38$, $r = 0.30$, and $r = 0.41$). This suggests that the participant's subjective mental load have decreased by using GO-Finder.

Functionality of interface Figure 4.13 shows the results of questions Q1–Q7. In Q1–Q5, participants reported positive impressions with the proposed system. The Wilcoxon signed-rank test revealed a significant difference between the frame-based aid and object-based aid conditions in Q6 ($p = 0.007$) but not in Q7 ($p = 0.065$). However, Q6 and Q7 showed large and medium effect sizes ($r = 0.55$, and $r = 0.38$), respectively, suggesting that GO-Finder was more useful in finding object locations compared to under the frame-based aid condition.

Comfort on neck-mounted camera The participants reported slightly positive feedback on average regarding comfort of camera (mean and 95% CI: 4.6 ± 1.2). Some preferred attaching cameras their glasses instead of their necks.

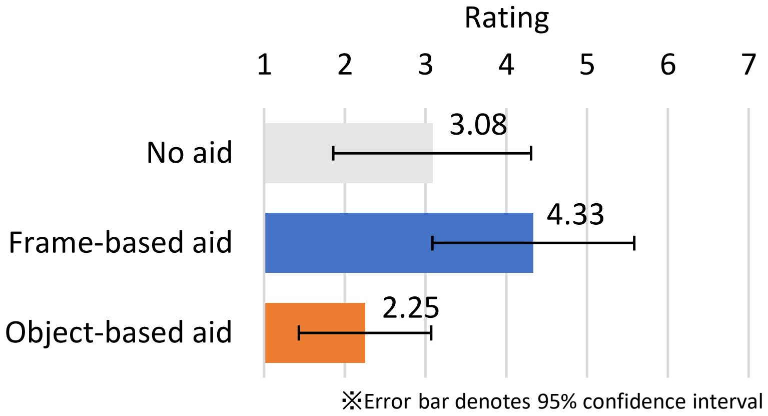


Figure 4.12: Ease of task (easy=1, difficult=7).

Table 4.7: Result of Wilcoxon signed-rank tests on ease of task.

			95% CI		Effect size
	Z	p	LB	UB	r
No aid/frame-based aid	-1.857	0.063	-2.581	0.081	**0.38**
No aid/object-based aid	-1.501	0.133	-0.596	2.263	**0.30**
Frame-based aid/object-based aid	-2.027	**0.043**	0.540	3.627	**0.41**

4.6.2 Usability Test

In addition to the main result, we asked the participants to answer the System usability scale (SUS) test [171]. Table 4.8 summarizes the SUS scores of each participant. The average score and its 95% confidence interval among all the participants were 75.4 ± 8.6. Based on acceptable ranges [172], 8 out of 12 participants evaluated GO-Finder as acceptable while one participant (P04) evaluated it as unacceptable. Low-scored participants mainly pointed out the difficulty in browsing the object timeline (see 4.6.2).

Observation and Feedback

Video observation In general, participants first looked for objects that they remembered and used the system when they were not confident with the location. When using the system, they looked for thumbnails showing the

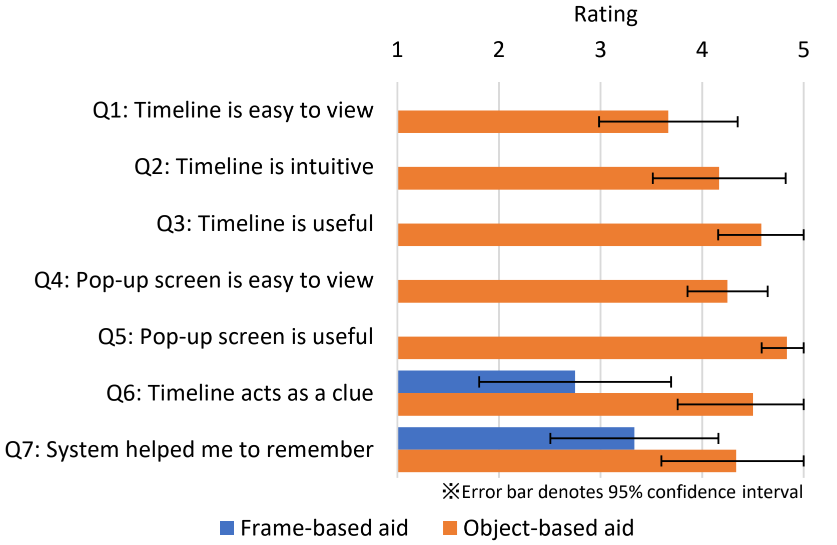


Figure 4.13: Results of questionnaire.

target object, inferred the location from the pop-up screen, and successfully retrieved the object. Two users persist using GO-Finder even when the system failed to discover the target objects (P06 and P10).

Usefulness of object-based aid Eleven out of the 12 participants said that GO-Finder was convenient to use. Only with a brief instruction, they were able to retrieve the forgotten locations with GO-Finder. They preferred the intuitiveness of the hand-held object timeline:

> A01: *The stuffs I wanted was displayed on the timeline. The system helped me because once pushed the icon it also displayed the location of them.* (P07)

> A02: *The function which I want most was there. Because the objects were highlighted and zoomed in, I could notice the target objects and retrieve the last moment by tapping the thumbnail.* (P08)

They felt more secure and confident at retrieval:

Table 4.8: Result of SUS test.

ID	Gender	SUS score (rank)
P01	Female	60 (D)
P02	Male	65 (C)
P03	Female	80 (A)
P04	Male	47.5 (F)
P06	Male	85 (C)
P07	Male	77.5 (B+)
P08	Male	77.5 (B+)
P09	Male	77.5 (B+)
P10	Male	87.5 (A+)
P11	Male	100 (A+)
P12	Male	70 (C)
P13	Male	77.5 (B+)

> A03: *Since I don't have to rely on my intuition, I looked at the smartphone once I felt lost. By using the system, I often felt confident about the location.* (P10)

A few participants trusted the system's output rather than their memory:

> A04: *I arrived at the wrong location since I relied on the system. I didn't remember my memory but inferred the location from the pop-up screen and got wrong.* (P13)

Comparison to frame-based aid In contrast, nine participants gave negative feedback regarding the frame-based aid condition. They mainly complained that the timeline often did not capture the exact moment of leaving objects. Difficulty in finding critical scenes from large field-of-view images was also reported:

> A05: *Since the images often don't capture the scene when holding objects, I found myself zooming into the image but found nothing for several times.* (P10)

> A06: *At a glance, all the thumbnails looked almost the same. I found difference when I looked into their details.* (P04)

The user has to additionally remember how they left the objects during the arrangement, sometimes being confused by their behavior:

> A07: *I was deluded by myself attempting to leave the object once but actually done it afterward.* (P03)

> A08: *I didn't find difference between searching without any aid and using the frame-based aid. I opened all the thumbnails because I had no idea.* (P02)

One participant preferred the frame-based timeline because thumbnails were evenly sampled in chronological order:

> A09: *I preferred that (the frame-based timeline) because the entire timeline was available and I could infer how I searched by looking at an image and the image next to it.* (P06)

On interface of system Participants preferred thumbnail images given from their point of view:

> A10: *The objects were shown by image, and were taken when I lost the object. The system was convenient since critical moments were captured in the timeline.* (P09)

While participants gave positive feedback for every component of the interface (Q1–Q7), they gave lower scores on the ease of using the hand-held object timeline (Q1). First, over-segmentation of the objects confused some participants:

> A11: *I found about four thumbnails showing a tennis ball. I had no idea which one to press[...].* (P01)

The quality of thumbnail images (brightness, occlusion, contrast, and viewpoint) also made it difficult for the participants to find the object of interest:

> A12: *[...]the thumbnail image of the last scene was difficult to identify. For example, I had to zoom in (to the thumbnail) when I looked for the pouch.* (P02)

Privacy concerns While we expected to have negative feedback on capturing images, six participants reported that they were not concerned with recording videos while three participants raised specific concerns:

> A13: *I don't feel any discomfort since I know what the system does; maybe because I know the system only collects information on objects. It might be different if the system captures people's faces.* (P12)

> A14: *I hesitate to wear this camera because I don't like myself being kept under surveillance. It's just my feeling, not a logical consequence.* (P09)

Some participants changed their behavior since they were aware of being recorded even though we did not give them any warning:

> A15: *I thought it was better not to hide the camera.* (P02)

Suggestions on improvement Two participants suggested playing a video snippet instead of a static image on the pop-up screen:

> A16: *I think it'll be easier to remind if I can view the before and after of the last scene. I don't think people can remind only from a static image.* (P09)

Regarding real-world use, participants suggested querying by background (P01, 04) and time (P12). They stated that the object itself is not a key to remember a scene and requested a functionality to filter the candidates by their own.

4.6.3 Results of Study II

We collected 16 trials per interface for each user, 96 trials per interface as a total.

Table 4.9: Retrieval time per item (sec).

Condition	Mean	95% CI	Min	Max
Baseline (object view only)	26.9	21.4–32.3	5.3	176.1
Full (object + scene + time + rec.)	35.7	28.5–43.0	5.1	281.5

Retrieval Time

Table 4.9 reports the mean retrieval time per item across all the participants and items. The unpaired t-test did not reveal a significant difference between the two conditions ($p = 0.084$). Considering the small effect size ($d = 0.25$) and the large variance among objects, the difference was not clear under this condition. We observed high variance among objects—some objects were easily identified in around 15 seconds while some other objects were difficult to find and took more than one minute on average. In the worst case, the participants took nearly five minutes to find an object, scrolling back and forth but cannot find it for a long time. While objects with a distinctive color (*e.g.*, blue stapler) were easy to find, less-textured, common (*e.g.*, mug and towel), or very unique (*e.g.*, thermometer) objects were more difficult to find.

We think that the increased retrieval time was brought about by additional page transition between different views and hierarchies. Practically, the ten-second difference on average might have affected the entire user experience negatively using the full interface.

Questionnaire

We observed a clear preference in which features they frequently used. Table 4.10 and Figure 4.14 report the most frequently used features and the usability ratings of each feature. We resorted to subjective evaluation since multiple features are sometimes used in one trial. We omitted six responses from P15 from the rating since they did not use the time view and similar object recommendation.

First of all, even in a longer list length, the object view got a positive

Table 4.10: Most frequently used features among participants.

Participant	Most used	Second most	Third most
P14	Object	Scene	Recommendation
P15	Object	Scene	(Unused)
P16	Object	Scene	Recommendation
P17	Scene	Object	Recommendation
P18	Object	Time	Recommendation
P19	Object	Recommendation	Time

rating. Five out of six participants used this functionality as a primary way to search the target objects. On the other hand, the rating of other features varied among participants. The scene view got a positive rating on usefulness by four out of six participants while one complained about the visibility and intuitiveness of the interface. The time view was less frequently used compared to the scene view but got a positive impression on usefulness by three participants. The recommendation was relatively used by two participants both reporting positive results on usefulness while the other participants did not use it or got a negative impression on using it. Among those three features other than object view, four participants used the scene view, one used the time view, and one used the recommendation most often.

Participant Feedback

We received mixed feedback on how they used the newly introduced features.

Retrieval strategy

In the baseline condition (object view only), the participants were allowed to do only one thing: they simply searched by scrolling down the object view. The participants reported using their own memory on locations and time to navigate through the timeline although no support was provided from the interface.

> A17: *If I remembered the place and time I saw the object, I searched the object using those cues. Otherwise, I scrolled through*

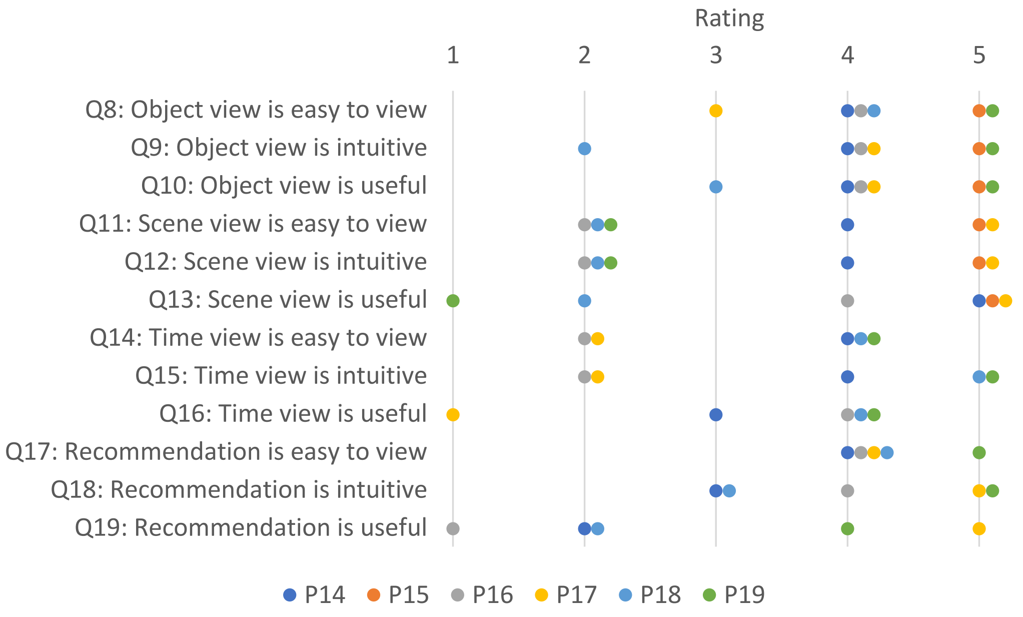


Figure 4.14: Results of questionnaire.

> *the list looking for the color of the object as a cue.* (P16)

Two participants reported that they relied on a simple visual search.

> A18: *I reflectively looked for the images (using the object view).* (P14)

> A19: *I looked for objects from a distance. It was easy to find the ones with distinctive colors.* (P19)

In the full condition, the participants used the newly introduced features as a supplement of the object view.

> A20: *Since there are many types of objects and it seems that it will take time to search in the object view, I mainly used the scene view based on the memory of the background of the object and the place when I saw the object. When I didn't remember the background, I searched for a similar-looking item using the object view and then used the recommendation screen (to jump to the target).* (P17)

> A21: *I searched for objects that I remembered in association with the event in the time view. For objects with a distinctive color, I used the object view.* (P18)

Scene view
The scene view was used as we expected.

> A22: *(By using the scene view,) I basically looked at the place where the object I was looking for was likely to appear.* (P17)

Some participants complained about the uncertainty on which scene to select.

> A23: *I was annoyed when I made a memory mistake in the scene I was counting on (and couldn't find the object).* (P14)

> A24: *The scene view was quite difficult to use. It was further divided into several scenes in the place I want to look for, so I couldn't decide which scene to look in.* (P18)

Time view
Although the time view was less frequently used, one participant used the time view rather than the scene view:

> A25: *I was looking for it [the object] by guessing the event when it appeared.* (P18)

However, similar to the scene view, it was harmful if the participant's guess was wrong:

> A26: *There were several times when I thought it [the object] was within in this time (window) but actually it wasn't.* (P18)

Recommendation
Two participants reported successful attempts using the recommendation.

> A27: *(When I wanted to find the oyster sauce) I found a seasoning in the object view, so I tapped on it. Then chili oil or something came up on the recommendation screen, so I tapped again then the thing I wanted has appeared.* (P17)

However, the current implementation could not accommodate if the user wanted to recommend a similar type of object.

> A28: *For example, when I had two mugs, I thought if I chose the first one, the system would offer me the other one, but it didn't recommend it.* (P14)

> A29: *When I was looking for white scissors, I was disappointed that the system didn't recommend it from the green scissors.* (P16)

Comparison between baseline and full interface
Three participants preferred using the full interface overall despite the worse retrieval time observed in this study. The participants were able to found their useful situations using the additional features. They expected the newly introduced features to be more effective in everyday situations.

> A30: *Considering that the number of images will dramatically increase when it comes to real-world situations, it would be great if the images could be separated by scene or by time, and in such cases, I would like to use the full interface.* (P18)

The other participants pointed out the additional cost to decide which feature to use:

> A31: *I wasn't sure which features to use in the full interface condition.* (P15)

On the other hand, the baseline interface was basically preferred of its simplicity. Three participants felt the object view is enough because what to do is clear.

> A32: *(The baseline system was) relatively simple because you can intuitively go down the list and find the one you want.* (P14)

One participant preferred the baseline interface because they are used to browse a large number of images using smartphones:

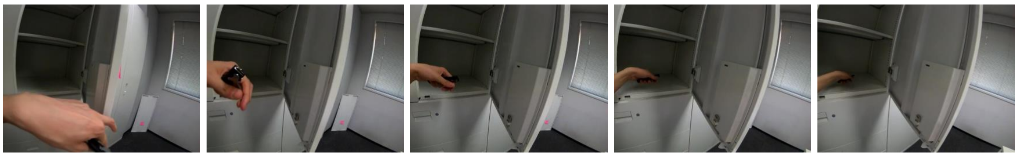


Figure 4.15: System failure under severe occlusion.

> A33: *I think it's fine for the generation that has a habit of scrolling through photos on their smartphones. On the other hand, if my parents' generation were to look at their smartphones together and scroll, they would ask me to stop.* (P16)

However, it was also clear that the object view alone is helpless when the participant has no idea when the desired item appeared:

> A34: *I was stressed when I looked (the list) from top to bottom but can't find what I was looking for.* (P19)

Suggestions on improvement
Four participants suggested that showing textual information such as event (P14 and P16), time (P14 and P17), and location name (P16) would help the retrieval.

> A35: *I think using speech is quicker if the system can recognize the object type.* (P14)

4.7 Discussion

4.7.1 Usefulness of GO-Finder

In the first user study, we confirmed that GO-Finder enabled the participants to retrieve hidden objects with less mental load. Quantitative and qualitative feedback suggests that the users gained confidence by immediately accessing the last moment of when the objects were seen. The frame-based timeline showing uniformly sampled video frames was not effective in this task. Since the object retrieval task should be solved as quickly as possible, users requested direct access to the location rather than having to keep relying on their memory.

System evaluation showed that the hand-held object discovery algorithm of GO-Finder successfully extracted hand-held object instances while efficiently excluding other unrelated objects. GO-Finder worked without explicit registration, making it easy for users to start using it.

Regarding the idea of using object images as a query, participants adapted quickly to browsing objects using the hand-held object timeline (Q3). The timeline shown as a list of images was evaluated as intuitive and participants were able to access the object of interest immediately (A01, A02, and A10). Since the thumbnail images are captured from almost the same view as the participant's one, this timeline also worked as a clue to remembering (Q6).

4.7.2 Interface Design

In the second user study, we conducted a usability study on the GO-Finder's smartphone interface on a longer video sequence of 75 minutes. Based on the quantitative result and user feedback, we summarize the findings as follows:

Scrolling the image timeline was more positively accepted than expected: As shown in the mean retrieval time and user feedback (A18, A19, A32, and A33), it turned out that scrolling a long list was better accepted without difficulty than we expect. Our result matches the observation that people using smartphones preferred to browse and scroll their way directly to the desired information [173]. Their work also reported participants using visual reminders such as screenshots to navigate to the desired item, also confirmed in our study (A19). However, several participants admitted that the situation may change if more objects appeared, or used the interface after enough time has passed since the last time they saw the objects. User evaluation in a more realistic situation is required to further support this finding.

Filtering by scene or time is a double-edged sword: We confirmed that users prefer filtering by scene or time when they can recollect the rough location or time where they once saw the object (A17, A22, and A25). However, due to the additional page transition introduced, it did not show improvement in the retrieval time. Since we aim to create a fully automatic system, we did not include any textual information in the interface. Taking feedback (P14, P16, and P17) into account, adding the side information of place names, time, and object types to the interface, using external information from GPS, electronic map, or user input, may help users rapidly filter the context.

Proactive recommendations may reduce the scrolling effort: Although not fully accepted by the participants, we found that recommendation could be used as a better efficient way instead of scrolling since the user does not need to remind the context but only find similar-looking items (A27-29). By providing multiple options on similarity metrics (*e.g.*, color, shape, and object type) followed by enough training, we may be able to provide a better efficient retrieval interface.

4.7.3 Privacy Issues

The use of wearable cameras raises privacy concerns in real-world use [174, 175]. Although GO-Finder's contents are not shared among other users, the privacy and comfort of bystanders must be secured. One way out of the difficulty is to filter out sensitive contents while storing only the information relevant to hand-held objects. Since GO-Finder only requires the last scene of an object, other frames are no longer needed as we store the feature vector of object detections. Last scenes will be updated as objects re-appears so images would not be kept stored permanently. Additionally, we can remove identity information by running an off-the-shelf face detector since we do not need bystander's information. Supported by the positive comments, we believe that GO-Finder can be used with minimal interference.

4.7.4 Sensitivity against Hyperparameters

Changing the hyperparameters (similarity thresholds) of the object (scene) discovery algorithm may affect the user experience while we fixed them throughout the study. As the algorithm will not achieve 100% accuracy, a proactive mechanism to ask whether the association is correct or not—something similar to facial recognition confirmation introduced in Google Photos—could be one solution.

4.7.5 Limitations

Object Re-identification

The proposed system failed to discover objects under severe occlusion by hands (*e.g.*, waiter's corkscrew, see Figure 4.15). In this example, the participant gripped the corkscrew so that it was severely occluded by the hand. This confused both the detector and clustering algorithm in determining the correct bounding box and appearance feature to identify the object, resulting in over or under segmentation. As reported in the system evaluation, small objects tend to be occluded and would be problematic regarding real-world use. One potential solution is to use the object's unoccluded appearance outside the interaction. In addition, when an object is occluded by hands during manipulation, it is natural to expect that the object would not disappear during the occlusion thus we can track the occluded object by tracking the hands. Although this problem was mitigated by metric learning (Subsection 4.4.3), a more sophisticated occlusion-aware re-identification mechanism also robust to viewpoint change is demanded for further improvement.

Evaluation in Everyday Life

In the second study, we investigated the usability of GO-Finder on much longer sequences with around 400 object clusters. Although we confirmed that users can browse through many objects with the help of filtering and recommendation, the usability in the daily field is not yet investigated. In practice, we often misplace objects after few days or months after their last appearance. Since this situation is almost impossible to reproduce in a controlled setting, a long-term study connected with the user's daily context is demanded. Towards this direction, additional challenges such as removal of uninterested items and usability on a dynamically updated timeline would appear and must be resolved.

Multi-User Scenarios

We assume each object is manipulated by a single user. However, in multi-user scenarios, we cannot track objects if they are moved by other people. One plausible solution is to share object information among users, which would be a trade-off between privacy protection.

4.8 Conclusion

In this chapter, I have presented GO-Finder, a registration-free wearable camera-based system for assisting users in finding lost objects. It supports the finding of an arbitrary number of objects based on two key ideas: hand-held object discovery and image-based candidate selection. Hand-held object detection and identification techniques were utilized to implement an automatic system that does not require any request from users. Furthermore, additional features of context-based candidate filtering are introduced to support efficient object retrieval in a realistic situation of a large number of objects appears. The first user study of performing a controlled object retrieval task revealed that by using GO-Finder users can find the location of lost objects correctly with a reduced mental load. Even the objects were registered automatically without user intervention, the participants were able to identify the target object using the image-based hand-held object timeline. The second user study of searching for objects that appeared in a longer video suggested the usefulness of the context-based candidate filtering against longer sequences while the simple hand-held object timeline was also shown to be more effective than expected. Going beyond tracking only a small number of selected objects, GO-Finder could be used as a practical tool to help find various unexpectedly lost objects in daily life. Future work includes long-term (*e.g.*, few weeks) evaluations on naturalistic situations of losing objects.

Chapter 5

Conclusion and Future Work

5.1 Summary

In this thesis, I have presented methods for recognizing when and for which object the hand-object interaction occurs that can be generalized to unknown, dynamic, and cluttered scenes in the real world. In addition, I have applied the above techniques to assistive technologies and have proposed a practical system for assisting users in finding lost objects. In Chapter 2, a video-based method to predict contact states between hands and objects was introduced. The semi-supervised method based on motion-based pseudo-labeling and guided label correction enabled the model to predict contacts between hands and objects from a small amount of training data. In Chapter 3, a large-scale and challenging benchmark on identifying object instances appearing in in-the-wild videos was proposed. It was shown that the model trained by our dataset exhibits better robustness against dynamic changes in appearance compared to the model trained by product images and large-scale image databases. As a result, a practical scheme to associate re-appearing object instances in real-world, dynamic environments was established. In Chapter 4, a system that assists users in finding lost hand-held objects by automatically detecting and tracking hand-held objects appearing in first-person videos was introduced. User studies showed that the fully-automatic hand-held object discovery algorithm enables users to find various unexpectedly lost objects in daily life.

These works were systematically designed through the unified concept of hand-object interaction mining, which aims to develop methods that generalize to novel scenes without assuming target environments or objects, in an unsupervised and scalable manner. Taken as a whole, the methods, datasets, and systems proposed in this thesis offer a scalable and practical scheme for recognizing and understanding long-term hand-object interactions in the real world.

This work's contributions are summarized as follows:

- Contributed a scalable approach of predicting contacts between hands and objects, which actively extracts useful knowledge from motion-based pseudo-labels through label correction.
- Contributed a new dataset and benchmark on category-agnostic object identification in dynamic real-world environments, whereas previous work studied object instance identification is mostly static scenes.
- Contributed the first registration-free system of assisting users in finding arbitrary lost hand-held objects through a wearable camera.

5.2 Future Directions

5.2.1 State-Aware Modeling of Objects

In this thesis, the fundamental problems in recognizing object contact state change (Chapter 2) and object instance identification in dynamic environments (Chapter 3) were studied. The above works can be interpreted as parts of the problem of recognizing current object states from visual input.

Humans perform actions to cause changes to the attributes of an object or person. Therefore, recognizing what kinds of attributes exist and their current state can be a powerful clue to correctly recognize the intent of human activities. Although this object state recognition task has been studied in simple manipulation tasks [126, 176, 177], the object's state-space was defined in advance and typically limited to binary states. However, object states will be not limited to simple binary states (*e.g.*, open/close, on/off) and their state space would also change by the user's context. For example, when serving several cups of coffee from a pot, the states of a pot should be defined more than a binary state (filled or empty) to serve coffee fairly to the cups. Beyond recognizing the object's location, discovering and recognizing its internal states will be one promising direction to realize a fine-grained understanding of human activities. For this task as well, since it is unrealistic to define and annotate all the patterns in advance, a data-driven, unsupervised approach (*e.g.*, [178]) will be promising.

5.2.2 Learning from Narrative

This thesis has contributed to the smallest fundamental elements for understanding hand-object interaction, that is, the methods for recognizing spatial and temporal relationships between hands and objects. Low-level cues such as motion direction and image intensity were used for supervision. However, the semantic aspect (what objects/actions are occurring) of hand-object interaction was completely ignored. Each human action will be equipped with his or her intent. Towards a practical application, it is also important to recognize its semantic aspects such as manipulation action, object categories and their goal (*e.g.*, pour water from a pitcher to fill the cup and drink) in understanding hand-object interactions.

While these semantic understanding tasks have been studied by providing discrete labels for supervision, it is clear that the amount of annotation will reach a plateau as the actions become more complicated. To overcome this issue, natural language data was found to be used as a rich knowledge base in connecting video and language. For example, audio transcripts included in instructional videos are used to learn a joint text-video embedding [179]. Audio transcripts describing the actions in videos are efficiently collected in recent first-person vision datasets [23, 180] and used for action retrieval [181]. Notably, the Ego4D benchmark [180] composed of 3,400 hours of first-person videos with various activities was recently released with temporally dense narrations for all the videos, producing 3.85 million sentences in total. These natural language resources have the potential to further understand hand-object interactions from video. At the moment, the above resources are used mostly to acquire video-level representation [179, 181, 182] and only a few works aim to learn object-level embeddings [183, 184]. Learning a hand and object-centric embedding for understanding hand manipulation and object state change will be a promising direction.

5.2.3 Environment-Aware Interactive System for Assisting User Activities

The GO-Finder system proposed in Chapter 4 was designed to automatically collect users' surrounding context on their manipulated objects. This idea can be further extended to collect a broader range of information on the user's context in real-time to assist users in assembly task [9, 185] and assistive livings [10]. For example, we may think of a system of discovering on/off switches in the room and tracking their state using a wearable camera, instead of placing external markers in advance [153]. Such systems can be naturally combined with augmented reality, which directly overlays necessary information to the user's workspace.

In addition, GO-Finder was intentionally designed not to ask any inquiry on using the system to not impose any burden on the user. However, it also has the disadvantage that it cannot always provide information in the form desired by the user. Towards an intelligent system to assist one's life, an interactive system that gives users the option to correct the system's wrong or undesirable information will be also a promising direction.

Bibliography

[1] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. *Technical Report TR 01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN*, 2001.

[2] Eadom Dessalene, Chinmaya Devaraj, Michael Maynord, Cornelia Fermuller, and Yiannis Aloimonos. Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, 2021.

[3] Ivan Rodin, Antonino Furnari, Dimitrios Mavroeidis, and Giovanni Maria Farinella. Predicting the future from first person (egocentric) vision: A survey. *Computer Vision and Image Understanding*, 211:103252, 2021.

[4] Jirapat Likitlersuang, Elizabeth R Sumitro, Tianshi Cao, Ryan J Visée, Sukhvinder Kalsi-Ryan, and José Zariffa. Egocentric video: a new tool for capturing hand use of individuals with spinal cord injury at home. *Journal of Neuroengineering and Rehabilitation*, 16(1):83, 2019.

[5] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 6169–6176, 2021.

[6] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics*, 39(4):87–1, 2020.

[7] Takeo Kanade and Martial Hebert. First-person vision. *Proceedings of the IEEE*, 100(8):2442–2453, 2012.

[8] Steve Mann. Wearable computing: A first step toward personal imaging. *Computer*, 30(2):25–32, 1997.

[9] Dima Damen, Teesid Leelasawassuk, and Walterio Mayol-Cuevas. You-do, i-learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance. *Computer Vision and Image Understanding*, 149:98–112, 2016.

[10] Marco Leo, G Medioni, M Trivedi, Takeo Kanade, and Giovanni Maria Farinella. Computer vision for assistive technologies. *Computer Vision and Image Understanding*, 154:1–15, 2017.

[11] Andrea Bandini and José Zariffa. Analysis of the hands in egocentric vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, 2020.

[12] Rui Li, Zhenyu Liu, and Jianrong Tan. A survey on 3d hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition*, 93:251–272, 2019.

[13] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019.

[14] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. *Proceedings of*

the IEEE International Conference on Computer Vision, pages 12417–12426, 2021.

[15] Yana Hasson, Gül Varol, Ivan Laptev, and Cordelia Schmid. Towards unconstrained joint hand-object reconstruction from rgb videos. *Computing Research Repository, abs/2108.07044*, 2021.

[16] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–419, 2018.

[17] Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8709–8719, 2019.

[18] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *Proceedings of the European Conference on Computer Vision*, 2020.

[19] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3196–3206, 2020.

[20] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision*, pages 619–635, 2018.

[21] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7396–7404, June 2018.

[22] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9869–9878, 2020.

[23] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *International Journal of Computer Vision*, 2021.

[24] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems*, 2018.

[25] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, October 2019.

[26] T. Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3888–3897, 2019.

[27] Supreeth Narasimhaswamy, Trung Nguyen, and Minh Hoai. Detecting hands and recognizing physical contact in the wild. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, 2020.

[28] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision*, pages 720–736, 2018.

[29] Supreeth Narasimhaswamy and Saif Vazir. Workinghands: A hand-tool assembly dataset for image segmentation and activity mining. In *Proceedings of the British Machine Vision Conference*, page 258, 2019.

[30] Javier Romero, Hedvig Kjellström, and Danica Kragic. Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation*, pages 458–463, 2010.

[31] Dimitrios Tzionas and Juergen Gall. 3d object reconstruction from hand-object interactions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 729–737, 2015.

[32] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021.

[33] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5031–5041, 2020.

[34] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: a dataset of whole-body human grasping of objects. In *Proceedings of the European Conference on Computer Vision*, pages 581–600, 2020.

[35] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[36] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A Argyros, and Abderrahmane Kheddar. Hand-object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2883–2896, 2017.

[37] Shuichi Akizuki and Yoshimitsu Aoki. Tactile logging for understanding plausible tool use based on human demonstration. In *1st International Workshop on Vision for Interaction and Behaviour Understanding*, page 334, 2019.

[38] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2020.

[39] Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef Sivic. Estimating 3d motion and forces of person-object interactions from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8640–8649, 2019.

[40] Davis Rempe, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *Proceedings of the European Conference on Computer Vision*, pages 71–87, 2020.

[41] Jesse Scott, Bharadwaj Ravichandran, Christopher Funk, Robert T Collins, and Yanxi Liu. From image to stability: Learning dynamics from human pose. In *Proceedings of the European Conference on Computer Vision*, pages 536–554, 2020.

[42] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017.

[43] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. Learning instance segmentation by interaction. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[44] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label

noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[45] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Proceedings of Advances in Neural Information Processing Systems*, volume 31, 2018.

[46] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[47] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature dependent label noise: a progressive approach. In *Proceedings of the International Conference on Learning Representations*, 2021.

[48] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of International Conference on Machine Learning*, pages 2304–2313, 2018.

[49] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of Advances in Neural Information Processing Systems*, 2018.

[50] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proceedings of the International Conference on Learning Representations*, 2020.

[51] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11442–11450, 2021.

[52] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Proceedings of the IEEE International Conference on Image Processing*, pages 3464–3468, 2016.

[53] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019.

[54] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[55] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 438–451, 2010.

[56] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *Computing Research Repository, abs/1607.06450*, 2016.

[57] Eddy Ilg, Nikolaus Mayer, T Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462 – 2470, 2017.

[58] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

[59] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[60] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 36–52, 2016.

[61] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, International Conference on Machine Learning*, 2013.

[62] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[63] Michele Merler, Carolina Galleguillos, and Serge Belongie. Recognizing groceries in situ using in vitro training data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[64] Miguel Lagunes-Fortiz, Dima Damen, and Walterio Mayol-Cuevas. Learning discriminative embeddings for object recognition on-the-fly. In *Proceedings of the International Conference on Robotics and Automation*, pages 2932–2938, 2019.

[65] Kyungjun Lee and Hernisa Kacorri. Hands holding clues for object recognition in teachable machines. In *Proceedings of the ACM CHI Conference of Human Factors in Computing Systems*, pages 1–12, 2019.

[66] Takuma Yagi, Takumi Nishiyasu, Kunimasa Kawasaki, Moe Matsuki, and Yoichi Sato. Go-finder: A registration-free wearable system for assisting users in finding lost objects via hand-held object discovery. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 139–149, 2021.

[67] Sameer A Nene, Shree K Nayar, and Hiroshi Murase. Columbia object image library (coil-20). Technical report, Technical Report CUCS-005-96, Department of Computer Science, Columbia University, New York, 1996.

[68] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[69] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[70] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[71] Sultan Daud Khan and Habib Ullah. A survey of advances in vision-based vehicle re-identification. *Computer Vision and Image Understanding*, 182:50–63, 2019.

[72] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2020.

[73] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[74] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546, 2005.

[75] David Held, Sebastian Thrun, and Silvio Savarese. Robust single-view instance recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2152–2159, 2016.

[76] Giulia Pasquale, Carlo Ciliberto, Lorenzo Rosasco, and Lorenzo Natale. Object identification from few examples by improving the invariance of a deep convolutional neural network. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4904–4911, 2016.

[77] Miguel Lagunes-Fortiz, Dima Damen, and Walterio Mayol-Cuevas. Centroids triplet network and temporally-consistent embeddings for in-situ object recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10796–10802, 2020.

[78] Ziang Xie, Arjun Singh, Justin Uang, Karthik S Narayan, and Pieter Abbeel. Multimodal blending for high-accuracy instance recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2214–2221, 2013.

[79] Edward Hsiao and Martial Hebert. Occlusion reasoning for object detectionunder arbitrary viewpoint. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1803–1815, 2014.

[80] Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, and Jana Košecká. Multiview rgb-d dataset for object instance detection. In *Proceedings of the International Conference on 3D Vision*, pages 426–434, 2016.

[81] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, Oct 2017.

[82] Jean-Philippe Mercier, Mathieu Garon, Philippe Giguere, and Jean-François Lalonde. Deep template-based object instance detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1507–1516, 2021.

[83] Vaibhav Bansal, Stuart James, and Alessio Del Bue. re-obj: Jointly learning the foreground and background for object instance re-identification. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 402–413, 2019.

[84] Vaibhav Bansal, Gian Luca Foresti, and Niki Martinel. Where did i see it? object instance re-identification with attention. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 298–306, 2021.

[85] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.

[86] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the Conference on Robot Learning*, pages 17–26, 2017.

[87] Guofeng Zou, Guixia Fu, Xiang Peng, Yue Liu, Mingliang Gao, and Zheng Liu. Person re-identification based on metric learning: a survey. *Multimedia Tools and Applications*, pages 1–34, 2021.

[88] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016.

[89] Ruihang Chu, Yifan Sun, Yadong Li, Zheng Liu, Chi Zhang, and Yichen Wei. Vehicle re-identification with viewpoint-aware metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8282–8291, 2019.

[90] Bryan Ning Xia, Yuan Gong, Yizhe Zhang, and Christian Poellabauer. Second-order non-local attention networks for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3760–3769, 2019.

[91] Xiaofeng Ren and Matthai Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2009.

[92] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1817–1824. IEEE, 2011.

[93] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 509–516. IEEE, 2014.

[94] Xiaohan Wang, Tengyu Ma, James Ainooson, Seunghwan Cha, Xiaotian Wang, Azhar Molla, and Maithilee Kunda. The toybox dataset of egocentric visual object transformations. *The 4th Vision Meets Cognition Workshop at Computer Vision and Pattern Recognition*, 2018.

[95] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3750–3757, 2018.

[96] Yalong Bai, Yuxiang Chen, Wei Yu, Linfang Wang, and Wei Zhang. Products-10k: A large-scale product recognition dataset. *Computing Research Repository, abs/2008.10545*, 2020.

[97] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7822–7831, 2021.

[98] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d:

Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[99] Giulia Pasquale, Carlo Ciliberto, Francesca Odone, Lorenzo Rosasco, and Lorenzo Natale. Are we done with object recognition? the icub robot's perspective. *Robotics and Autonomous Systems*, 112:260–281, 2019.

[100] Shuang Wang and Shuqiang Jiang. Instre: a new benchmark for instance-level object retrieval and recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(3):1–21, 2015.

[101] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):289–303, 2017.

[102] Yichun Shi, Charles Otto, and Anil K Jain. Face clustering: representation and pairwise constraints. *IEEE Transactions on Information Forensics and Security*, 13(7):1626–1640, 2018.

[103] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1117–1125, 2019.

[104] Senhui Guo, Jing Xu, Dapeng Chen, Chao Zhang, Xiaogang Wang, and Rui Zhao. Density-aware feature embedding for face clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6698–6706, June 2020.

[105] Xuan-Bac Nguyen, Duc Toan Bui, Chi Nhan Duong, Tien D. Bui, and Khoa Luu. Clusformer: A transformer based clustering approach to unsupervised large-scale face and visual landmark recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10847–10856, June 2021.

[106] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13369–13378, June 2020.

[107] Vicky Kalogeiton and Andrew Zisserman. Constrained video face clustering using 1nn relations. In *Proceedings of the British Machine Vision Conference*, 2020.

[108] Yifan Xing, Tong He, Tianjun Xiao, Yongxin Wang, Yuanjun Xiong, Wei Xia, David Wipf, Zheng Zhang, and Stefano Soatto. Learning hierarchical graph neural networks for image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3467–3477, October 2021.

[109] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *Proceedings of the European Conference on Computer Vision*, September 2018.

[110] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[111] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[112] Godfrey N Lance and William Thomas Williams. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

[113] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973.

[114] Junfu Liu, Di Qiu, Pengfei Yan, and Xiaolin Wei. Learn to cluster faces via pairwise classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3845–3853, October 2021.

[115] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *Computing Research Repository, abs/2108.06753*, 2021.

[116] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[117] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[118] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *Proceedings of the British Machine Vision Conference*, 2019.

[119] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[120] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.

[121] Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[122] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

[123] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[124] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

[125] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[126] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2127–2136, 2017.

[127] Runtao Liu, Zhirong Wu, Stella Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, 2021.

[128] Rodney E Peters, Richard Pak, Gregory D Abowd, Arthur D Fisk, and Wendy A Rogers. Finding lost objects: Informing the design of ubiquitous computing services for the home. Technical Report GIT-GVU-04-01, Georgia Institute of Technology, 2004.

[129] Margery Eldridge, Abigail Sellen, and Debra Bekerian. Memory problems at work: Their range, frequency and severity. Technical Report EPC–92–129, Rank Xerox EUROPARC, 1992.

[130] David Elsweiler, Ian Ruthven, and Christopher Jones. Towards memory supporting personal information management tools. *Journal of the American Society for Information Science and Technology*, 58(7):924–946, 2007.

[131] Pixie Technology. The nation's biggest lost and found survey, by Pixie. https://tinyurl.com/yxrzbsnp, 2017. archived: 2017-12-06.

[132] Apple Inc. AirTag. https://www.apple.com/airtag/, 2021. accessed: 2021-08-06.

[133] Makoto Shinnishi. Hide and seek: Physical real artifacts which responds to the user. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, volume 4, pages 84–88, 1999.

[134] Tatsuyuki Kawamura, Tomohiro Fukuhara, Hideaki Takeda, Yasuyuki Kono, and Masatsugu Kidode. Ubiquitous memories: A memory externalization system using physical objects. *Personal and Ubiquitous Computing*, 11(4):287–298, 2007.

[135] Takahiro Ueoka, Tatsuyuki Kawamura, Yasuyuki Kono, and Masatsugu Kidode. I'm here!: A wearable object remembrance support system. In *Proceedings of the ACM International Conference on Mobile Human-Computer Interaction*, pages 422–427, 2003.

[136] Gaetano Borriello, Waylon Brunette, Matthew Hall, Carl Hartung, and Cameron Tangney. Reminding about tagged objects using passive rfids. In *Proceedings of the ACM International Conference on Ubiquitous Computing*, pages 36–53, 2004.

[137] Dan Xie, Tingxin Yan, Deepak Ganesan, and Allen Hanson. Design and implementation of a dual-camera wireless sensor network for object retrieval. In *Proceedings of the IEEE International Conference on Information Processing in Sensor Networks*, pages 469–480, 2008.

[138] Robert J Orr, Ronald Raymond, Joshua Berman, and A Fleming Seay. A system for finding frequently lost objects in the home. Technical Report GIT-GVU-99-24, Georgia Institute of Technology, 1999.

[139] Xiaotao Liu, Mark D Corner, and Prashant Shenoy. Ferret: Rfid localization for pervasive multimedia. In *Proceedings of the ACM International Conference on Ubiquitous Computing*, pages 422–440, 2006.

[140] Paul Wilson, Daniel Prashanth, and Hamid Aghajan. Utilizing rfid signaling scheme for localization of stationary objects and speed esti-

mation of mobile objects. In *Proceedings of the IEEE International Conference on RFID*, pages 94–99, 2007.

[141] Masaya Tanbo, Ryoma Nojiri, Yuusuke Kawakita, and Haruhisa Ichikawa. Active rfid attached object clustering method with new evaluation criterion for finding lost objects. *Mobile Information Systems*, 2017(3637814), 2017.

[142] Julie A Kientz, Shwetak N Patel, Arwa Z Tyebkhan, Brian Gane, Jennifer Wiley, and Gregory D Abowd. Where's my stuff? design and evaluation of a mobile system for locating lost items for the visually impaired. In *Proceedings of the ACM Conference on Computers and Accessibility*, pages 103–110, 2006.

[143] Ling Pei, Ruizhi Chen, Jingbin Liu, Tomi Tenhunen, Heidi Kuusniemi, and Yuwei Chen. Inquiry-based bluetooth indoor positioning via rssi probability distributions. In *Proceedings of the International Conference on Advances in Satellite and Space Communications*, pages 151–156, 2010.

[144] David Schwarz, Max Schwarz, Jörg Stückler, and Sven Behnke. Cosero, find my keys! object localization and retrieval using bluetooth low energy tags. In *Robot Soccer World Cup*, pages 195–206, 2014.

[145] Andreas Butz, Michael Schneider, and Mira Spassova. Searchlight–a lightweight search function for pervasive environments. In *Proceedings of the IEEE International Conference on Pervasive Computing*, pages 351–356, 2004.

[146] Markus Funk, Albrecht Schmidt, and Lars Erik Holmquist. Antonius: A mobile search engine for the physical world. In *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct*, pages 179–182, 2013.

[147] Markus Funk, Robin Boldt, Bastian Pfleging, Max Pfeiffer, Niels Henze, and Albrecht Schmidt. Representing indoor location of objects on wearable computers with head-mounted displays. In *Proceedings of the Augmented Human International Conference*, pages 1–4, 2014.

[148] Tile Inc. Find your keys, Wallet & phone with Tile's app and Bluetooth tracker device — Tile. https://www.thetileapp.com/en-eu/, 2017. archived: 2017-12-06.

[149] William F Brewer. Qualitative analysis of the recalls of randomly sampled autobiographical events. In MM Gruneberg, PE Morris, and RN Sykes, editors, *Practical Aspects of Memory: Current Research and Issues*, volume 1, pages 263–268. Wiley, 1988.

[150] Quan T Tran, Gina Calcaterra, and Elizabeth D Mynatt. Cook's collage. In *Proceedings of the International Conference on Home-Oriented Informatics and Telematics*, pages 15–32, 2005.

[151] Bernt Schiele, Nuria Oliver, Tony Jebara, and Alex Pentland. Dypers: Dynamic personal enhanced reality system. In *Proceedings of the International Conference on Computer Vision Systems*, 1999.

[152] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. Sensecam: A retrospective memory aid. In *Proceedings of the ACM International Conference on Ubiquitous Computing*, pages 177–193, 2006.

[153] Franklin Mingzhe Li, Di Laura Chen, Mingming Fan, and Khai N Truong. Fmt: A wearable camera-based object tracking memory aid for older adults. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–25, 2019.

[154] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353, 2012.

[155] Marc Bolaños and Petia Radeva. Ego-object discovery. *Computing Research Repository, abs/1504.01639*, 2015.

[156] Cristian Reyes, Eva Mohedano, Kevin McGuinness, Noel E O'Connor, and Xavier Giro-i Nieto. Where is my phone? personal object retrieval from egocentric images. In *Proceedings of the First Workshop on Lifelogging Tools and Applications*, pages 55–62, 2016.

[157] Gedas Bertasius, Hyun Soo Park, Stella X Yu, and Jianbo Shi. Unsupervised learning of important objects from first-person videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1956–1964, 2017.

[158] Cewu Lu, Renjie Liao, and Jiaya Jia. Personal object discovery in first-person videos. *IEEE Transactions on Image Processing*, 24(12):5789–5799, 2015.

[159] Keita Higuchi, Ryo Yonetani, and Yoichi Sato. Egoscanning: Quickly scanning first-person videos with egocentric elastic timelines. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 6536–6546, 2017.

[160] Irshad Abibouraguimane, Kakeru Hagihara, Keita Higuchi, Yuta Itoh, Yoichi Sato, Tetsu Hayashida, and Maki Sugimoto. Cosummary: adaptive fast-forwarding for surgical videos by detecting collaborative scenes using hand regions and gaze positions. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 580–590, 2019.

[161] Kyungjun Lee, Abhinav Shrivastava, and Hernisa Kacorri. Hand-priming in object localization for assistive egocentric vision. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 3422–3432, 2020.

[162] William Jones. Personal information management. *Annual Review of Information Science and Technology*, 41(1):453–504, 2007.

[163] Liadh Kelly, Yi Chen, Marguerite Fuller, and Gareth JF Jones. A study of remembered context for information access from personal digital archives. In *Proceedings of the International Symposium on Information Interaction in Context*, pages 44–50, 2008.

[164] J Indratmo and Julita Vassileva. A review of organizational structures of personal information management. *Journal of Digital Information*, 9(1), 2008.

[165] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 850–865, 2016.

[166] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[167] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[168] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[169] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[170] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[171] John Brooke. Sus: a "quick and dirty'usability. *Usability Evaluation in Industry*, page 189, 1996.

[172] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[173] Amalie Enshelm Jensen, Caroline Møller Jægerfelt, Sanne Francis, Birger Larsen, and Toine Bogers. I just scroll through my stuff until i find it or give up: A contextual inquiry of pim on private handheld devices. In *Proceedings of the Conference on Human Information Interaction & Retrieval*, pages 140–149, 2018.

[174] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2377–2386, 2014.

[175] Roberto Hoyle, Robert Templeman, Steven Armes, Denise Anthony, David Crandall, and Apu Kapadia. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 571–582, 2014.

[176] Yang Liu, Ping Wei, and Song-Chun Zhu. Jointly recognizing object fluents and tasks in egocentric videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2924–2932, 2017.

[177] Amy Fire and Song-Chun Zhu. Inferring hidden statuses and actions in video by causal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 48–56, 2017.

[178] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020.

[179] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019.

[180] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. *Computing Research Repository, abs/2110.07058*, 2021.

[181] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 450–459, 2019.

[182] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[183] Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video. *Advances in Neural Information Processing Systems*, 33:15133–15145, 2020.

[184] Reuben Tan, Bryan Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. *Advances in Neural Information Processing Systems*, 34, 2021.

[185] William Hoff and Hao Zhang. Learning object and state models for ar task guidance. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 272–273, 2016.

Publications

Publications Related to the Thesis

International Conferences and Workshops

[1] **Takuma Yagi**, Md. Tasnimul Hasan and Yoichi Sato. “Hand-Object Contact Prediction via Motion-Based Pseudo-Labeling and Guided Progressive Label Correction”. In *Proceedings of British Machine Vision Conference (BMVC)*, November 2021.

[2] **Takuma Yagi**, Takumi Nishiyasu, Moe Matsuki, Kunimasa Kawasaki and Yoichi Sato. “GO-Finder: A Registration-Free Wearable System for Assisting Users in Finding Lost Objects via Hand-Held Object Discovery”. In *Proceedings of ACM Intelligent User Interface (IUI)*, pages 139–149, April 2021.

Domestic Presentations (No peer review)

[3] 八木拓真, Md. Tasnimul Hasan, 佐藤洋一. 誘導付き逐次ラベル訂正に基づく映像からの手-物体接触判定. 第24回画像の認識・理解シンポジウム（MIRU2021）, 2021年7月

[4] 八木拓真, 西保匠, 川崎邦将, 松木萌, 佐藤洋一. GO-Finder: 手操作物体の発見に基づく事前登録不要のウェアラブル物探し支援システム. インタラクション, pages 739–744, 2021年3月.

Other Publications

International Journals

[5] Takehiko Ohkawa, **Takuma Yagi**, Atsushi Hashimoto, Yoshitaka Ushiku and Yoichi Sato. “Foreground-Aware Stylization and Consensus Pseudo-Labeling for Domain Adaptation of First-Person Hand Segmentation”. *IEEE Access*, volume 9, pages 94644–94655, June 2018.

International Conferences and Workshops

[6] **Takuma Yagi**, Karrtikeya Mangalam, Ryo Yonetani and Yoichi Sato. “Future Person Localization in First-Person Videos”. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 7593–7602, June 2018.

Domestic Presentations (No peer review)

[7] Takehiko Ohkawa, **Takuma Yagi**, Atsushi Hashimoto, Yoshitaka Ushiku and Yoichi Sato. “Foreground-Aware Stylization and Consensus Pseudo-Labeling for Domain Adaptation of First-Person Hand Segmentation”. 第24回画像の認識・理解シンポジウム（MIRU2021）, 2021年7月.

[8] 福嶋稜, 八木拓真, 馬場淳, 岩本拓也, 遠藤大介, 大澤正彦. 購買行動において認知的不協和を顕在化し解消を促進する窓エージェントの提案と検討. HAIシンポジウム, 2021年3月.

[9] Takehiko Ohkawa, **Takuma Yagi**, Atsushi Hashimoto, Yoshitaka Ushiku and Yoichi Sato. “Style Adapted DataBase: Generalizing Hand Segmentation via Semantics-aware Stylization”. IEICE Technical Report (PRMU2020), 2020年10月.

[10] 八木拓真, 西保匠, 川崎邦将, 松木萌, 佐藤洋一. 手操作物体の識別による手-物体インタラクション可視化システム. 第23回画像の認識・理解シンポジウム（MIRU2020）, 2020年8月.

[11] 八木拓真, 佐藤洋一. 運動情報を用いた手およびその接触物体の弱教師ありセグメンテーション. 第23回画像の認識・理解シンポジウム（MIRU2020）, 2020年8月.

[12] Donghao Wu, **Takuma Yagi**, Yusuke Matsui and Yoichi Sato. “Egocentric Pedestrian Motion Forecasting for Separately Modelling Pose and Location”. IEICE Technical Report (PRMU2019), 2020年3月.

[13] 八木拓真, 川崎邦将, 佐藤洋一. 周辺人物位置予測を行うウェアラブルシステム. 第22回画像の認識・理解シンポジウム（MIRU2019）, 2019年8月.

[14] 八木拓真, 品川政太朗, 秋山解, 加藤大貴, 島村僚, 又吉拓, 佐藤洋一. 【招待ショートサーベイ】ユーザ評価からみるHCI ～良いシステムの実現のためにCV研究者が学ぶこと～. 信学技報, volume 118, number 260, PRMU2018-67, pages 1–4, 2018年10月.

[15] 八木拓真, マンガラムカーティケヤ, 米谷竜, 佐藤洋一. 一人称視点映像における人物位置予測. 第21回画像の認識・理解シンポジウム（MIRU2018）, 2018年8月.

[16] 八木拓真, マンガラムカーティケヤ, 米谷竜, 佐藤洋一. 一人称視点映像における人物位置予測. 第211回CVIM研究会, 2018年3月.