

# The MECCANO Dataset: Understanding Human-Object Interactions from Egocentric Videos in an Industrial-like Domain

Francesco Ragusa  
IPLAB, University of Catania  
XGD-XENIA s.r.l., Acicastello, Catania, Italy  
francesco.ragusa@unict.it

Salvatore Livatino  
University of Hertfordshire  
s.livatino@herts.ac.uk

Antonino Furnari  
IPLAB, University of Catania  
furnari@dmi.unict.it

Giovanni Maria Farinella  
IPLAB, University of Catania  
gfarinella@dmi.unict.it

## Abstract

Wearable cameras allow to collect images and videos of humans interacting with the world. While human-object interactions have been thoroughly investigated in third person vision, the problem has been understudied in egocentric settings and in industrial scenarios. To fill this gap, we introduce MECCANO, the first dataset of egocentric videos to study human-object interactions in industrial-like settings. MECCANO has been acquired by 20 participants who were asked to build a motorbike model, for which they had to interact with tiny objects and tools. The dataset has been explicitly labeled for the task of recognizing human-object interactions from an egocentric perspective. Specifically, each interaction has been labeled both temporally (with action segments) and spatially (with active object bounding boxes). With the proposed dataset, we investigate four different tasks including 1) action recognition, 2) active object detection, 3) active object recognition and 4) egocentric human-object interaction detection, which is a revisited version of the standard human-object interaction detection task. Baseline results show that the MECCANO dataset is a challenging benchmark to study egocentric human-object interactions in industrial-like scenarios. We publicly release the dataset at <https://iplab.dmi.unict.it/MECCANO>.

## 1. Introduction

Being able to analyze human behavior from egocentric observations has many potential applications related to the recent development of wearable devices [1, 2, 3] which range from improving the personal safety of workers in a factory [10] to providing assistance to the visitors of a mu-

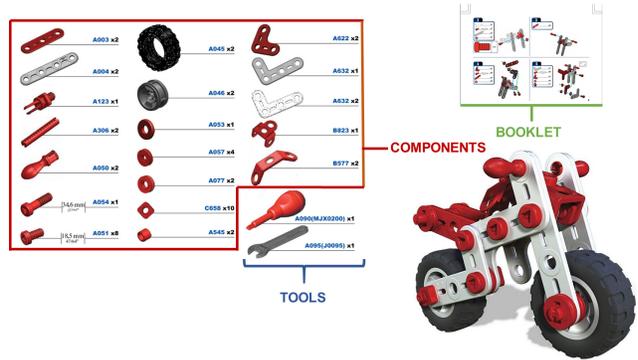


Figure 1. Toy model built by subjects interacting with 2 tools, 49 components and the instructions booklet. Better seen on screen.

seum [23, 50, 11]. In particular, with the rapid growth of interest in wearable devices in industrial scenarios, recognizing human-object interactions can be useful to prevent safety hazards, implement energy saving policies and issue notifications about actions that may be missed in a production pipeline [56].

In recent years, progress has been made in many research areas related to human behavior understanding, such as action recognition [24, 55, 26, 68, 35, 40], object detection [28, 27, 52, 51] and human-object interaction detection [29, 32, 53, 44]. These advances have been possible thanks to the availability of large-scale datasets [36, 41, 13, 38, 32, 8] which have been curated and often associated with dedicated challenges. In the egocentric vision domain, in particular, previous investigations have considered the contexts of kitchens [13, 38, 17], as well as daily living activity at home and in offices [47, 58, 16, 46]. While these contexts provide interesting test-beds to study user behavior in general, egocentric human-object

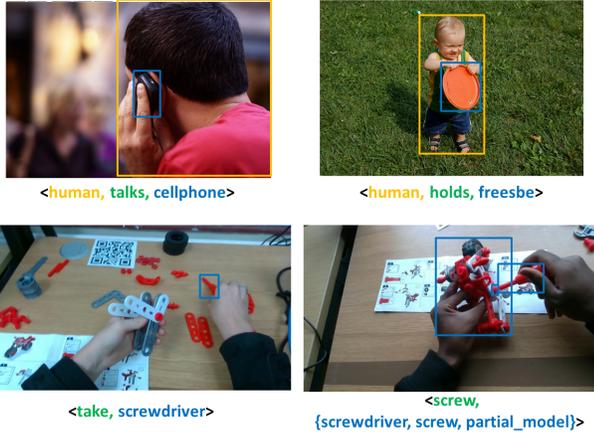


Figure 2. Examples of Human-Object Interactions in third person vision (first row) and first person vision (second row)<sup>2</sup>.

interactions have not been previously studied in industrial environments such as factories, building sites, mechanical workshops, etc. This is mainly due to the fact that data acquisition in industrial domains is difficult because of privacy issues and the need to protect industrial secrets.

In this paper, we present MECCANO, the first dataset of egocentric videos to study human-object interactions in industrial-like settings. To overcome the limitations related to data collection in industry, we resort to an industrial-like scenario in which subjects are asked to build a toy model of a motorbike using different components and tools (see Figure 1). Similarly to an industrial scenario, the subjects interact with tools such as a screwdriver and a wrench, as well as with tiny objects such as screws and bolts while executing a task involving sequential actions (e.g., take wrench, tighten bolt, put wrench). Despite the fact that this scenario is a simplification of what can be found in real industrial settings, it is still fairly complex, as our experiments show.

MECCANO was collected by 20 different participants in two countries (Italy and United Kingdom). We densely annotated the acquired videos with temporal labels to indicate the start and end times of each human-object interaction, and with bounding boxes around the active objects involved in the interactions. We hope that the proposed dataset will encourage research in this challenging domain. The dataset is publicly released at the following link: <https://iplab.dmi.unict.it/MECCANO>.

We show that the proposed dataset can be used to study four fundamental tasks related to the understanding of human-object interactions: 1) Action Recognition, 2) Active Object Detection, 3) Active Object Recognition and 4) Egocentric Human-Object Interaction Detection. While past works have already investigated the tasks of action recognition [13, 38, 42, 57], active object detection [47], and active object recognition [13] in the context of egocen-

tric vision, Human-Object Interaction (HOI) detection has been generally studied in the context of third person vision [32, 29, 48, 9, 69, 39, 64]. Since we believe that modelling actions both semantically and spatially is fundamental for egocentric vision applications, we instantiate the Human-Object Interaction detection task in the context of the proposed dataset.

HOI detection consists in detecting the occurrence of human-object interactions, localizing both the humans taking part in the action and the interacted objects. HOI detection also aims to understand the relationships between humans and objects, which is usually described with a verb. Possible examples of HOIs are “*talk on the cell phone*” or “*hold a frisbee*”. HOI detection models mostly consider one single object involved in the interaction [32, 31, 29, 67, 9]. Hence, an interaction is generally defined as a triplet in the form  $\langle human, verb, object \rangle$ , where the human is the subject of the action specified by a verb and an object. Sample images related to human-object interactions in a third-person scenario are shown in Figure 2-top. We define Egocentric Human-Object Interaction (EHOI) detection as the task of producing  $\langle verb, objects \rangle$  pairs describing the interaction observed from the egocentric point of view. Note that in EHOI, the human interacting with the objects is always the camera wearer, while one or more objects can be involved simultaneously in the interaction. The goal of EHOI detection is to infer the verb and noun classes, and to localize each active object involved in the interaction. Figure 2-bottom reports some examples of Egocentric Human-Object Interactions.

We perform experiments with baseline approaches to tackle the four considered tasks. Results suggest that the proposed dataset is a challenging benchmark for understanding egocentric human-object interactions in industrial-like settings. In sum, the contributions of this work are as follows: 1) we present MECCANO, a new challenging egocentric dataset to study human-object interactions in an industrial-like domain; 2) we instantiate the HOI definition in the context of egocentric vision (EHOI); 3) we show that the current state-of-the-art approaches achieve limited performance, which suggests that the proposed dataset is an interesting benchmark for studying egocentric human-object interactions in industrial-like domains.

## 2. Related Work

**Datasets for Human Behavior Understanding** Previous works have proposed datasets to tackle the task of Human-Object Interaction (HOI) detection. Among the most notable datasets, we can mention V-COCO [32], which adds 26 verb labels to the 80 objects of COCO [41],

<sup>2</sup>Images in the first row were taken from the COCO dataset [41] while those in the second row belong to the MECCANO dataset.

Dataset	Settings	EGO?	Video?	Tasks	Year	Frames	Sequences	AVG. video duration	Action classes	Object classes	Object BBs	Participants
MECCANO	Industrial-like	✓	✓	EHOI, AR, AOD, AOR	2020	299,376	20	20.79 min	61	20	64,349	20
EPIC-KITCHENS [13]	Kitchens	✓	✓	AR, AOR	2018	11.5M	432	7.64 min	125	352	454,255	32
EGTEA Gaze+ [38]	Kitchens	✓	✓	AR	2018	2.4M	86	0.05 min	106	0	0	32
THU-READ [58]	Daily activities	✓	✓	AR	2017	343,626	1920	7.44 min	40	0	0	8
ADL [47]	Daily activities	✓	✓	AR, AOR	2012	1.0M	20	30.0 min	32	42	137,780	20
CMU [17]	Kitchens	✓	✓	AR	2009	200,000	16	15.0 min	31	0	0	16
Something-Something [30]	General	X	✓	AR, HOI	2017	5.2 M	108,499	0.07 min	174	N/A	318,572	N/A
Kinetics [6]	General	X	✓	AR	2017	N/A	455,000	0.17 min	700	0	0	N/A
ActivityNet [20]	Daily activities	X	✓	AR	2015	91.6 M	19,994	2.55 min	200	N/A	N/A	N/A
HOI-A [39]	General	X	X	HOI, AOR	2020	38,668	N/A	N/A	10	11	60,438	N/A
HICO-DET [8]	General	X	X	HOI, AOR	2018	47,776	N/A	N/A	117	80	256,672	N/A
V-COCO [32]	General	X	X	HOI, OD	2015	10,346	N/A	N/A	26	80	N/A	N/A

Table 1. Comparative overview of relevant datasets. HOI: HOI Detection. EHOI: EHOI Detection. AR: Action Recognition. AOD: Active Object Detection. AOR: Active Object Recognition. OD: Object Detection.

HICO-Det [8], labeled with 117 verbs and 80 objects, HOI-A [39], which focuses on 10 verbs and 11 objects indicating actions dangerous while driving. Other works have proposed datasets for action recognition from video. Among these, ActivityNet [20] is a large-scale dataset composed of videos depicting 203 activities that are relevant to how humans spend their time in their daily lives, Kinetics [34, 6] is a dataset containing 700 human action classes, Something-Something [30] includes low-level concepts to represent simple everyday aspects of the world. Previous works also proposed datasets of egocentric videos. Among these datasets, EPIC-Kitchens [13, 15, 14] focuses on unscripted activities in kitchens, EGTEA Gaze+ [38] contains videos paired with gaze information collected from participants cooking different recipes in a kitchen, CMU [17] is a multi-modal dataset of egocentric videos including RGB, audio and motion capture information, ADL [47] contains egocentric videos of subjects performing daily activities, THU-READ [58] contains RGB-D videos of subjects performing daily-life actions in different scenarios. Table 1 compares the aforementioned datasets with respect to the proposed dataset. MECCANO is the first dataset of egocentric videos collected in an industrial-like domain and annotated to perform EHOI Detection. It is worth noting that previous egocentric datasets have considered scenarios related to kitchens, offices, and daily-life activities and that they have generally tackled the action recognition task rather than EHOI detection.

**Action Recognition** Action recognition from video has been thoroughly investigated especially in the third person vision domain. Classic works [37, 12] relied on motion-based features such as optical flow and space-time features. Early deep learning works fused processing of RGB and optical flow features with two-stream networks [55, 26, 63], 3D ConvNets are commonly used to encode both spatial and temporal dimensions [59, 60, 7], long-term filtering and pooling has focused on representing actions considering their full temporal extent [62, 26, 63, 68, 68, 40]. Other works separately factor convolutions into separate 2D spatial and 1D temporal filters [25, 61, 66, 49]. Among recent works, Slow-Fast networks [24] avoid using pre-

computed optical flow and encodes motion into a “fast” pathway (which operates at a high frame rate) and simultaneously a “slow” pathway which captures semantics (operating at a low frame rate). We assess the performance of state-of-the-art action recognition methods on the proposed dataset considering SlowFast networks [24], I3D [7] and 2D CNNs as baselines.

**HOI Detection** Previous works have investigated HOI detection mainly from a third person vision point of view.

The authors of [32] proposed a method to detect people performing actions able to localize the objects involved in the interactions on still images. The authors of [29] proposed a human-centric approach based on a three-branch architecture (InteractNet) instantiated according to the definition of HOI in terms of a <human, verb, object> triplet. Some works [48, 9, 69] explored HOI detection using graph convolutional neural networks after detecting humans and objects in the scene. Recent works [39, 64] represented the relationship between both humans and objects as the intermediate point which connects the center of the human and object bounding boxes. The aforementioned works addressed the problem of HOI detection in the third person vision domain. In this work, we look at the task of HOI detection from an egocentric perspective considering the proposed MECCANO dataset.

**EHOI Detection** EHOI detection is understudied due to the limited availability of egocentric datasets labelled for this task. While some previous datasets such as EPIC-KITCHENS [13, 14] and ADL [47] have been labeled with bounding box annotations, these datasets have not been explicitly labeled for the EHOI detection task indicating relationships between labeled objects and actions, hence preventing the development of EHOI detection approaches. Some related studies have modeled the relations between entities for interaction recognition as object affordances [43, 44, 22]. Other works tackled tasks related to EHOI detection proposing hand-centric methods [5, 4, 53]. Despite these related works have considered human-object interaction from an egocentric point of view, the EHOI detection task has not yet been defined or studied systematically in past works. With this paper we aim at providing a



Figure 3. Examples of data acquired by the 20 different participants in two countries (Italy, United Kingdom).

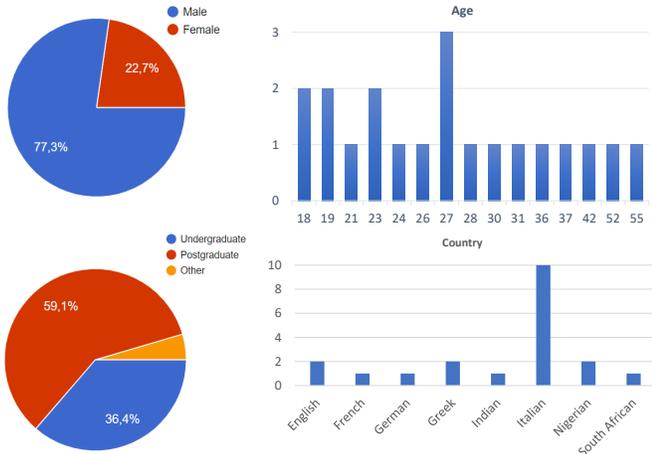


Figure 4. Statistics of the 20 participants.

definition of the task, a suitable benchmark dataset, as well as an initial evaluation of baseline approaches.

### 3. MECCANO Dataset

#### 3.1. Data Collection

The MECCANO dataset has been acquired in an industrial-like scenario in which subjects built a toy model of a motorbike (see Figure 1). The motorbike is composed of 49 components with different shapes and sizes belonging to 19 classes. In our settings, the components *A054* and *A051* of Figure 1 have been grouped under the “screw” class, whereas *A053*, *A057* and *A077* have been grouped under the “washers” class. As a result, we have 16 component classes<sup>3</sup>. Note that multiple instances of each class are necessary to build the model. In addition, 2 tools, a *screwdriver* and a *wrench*, are available to facilitate the assembly of the toy model. The subjects can use the instruction booklet to understand how to build the toy model following the sequential instructions.

For the data collection, the 49 components related to the considered 16 classes, the 2 tools and the instruction booklet have been placed on a table to simulate an industrial-like en-

<sup>3</sup>See supplementary material for more details.

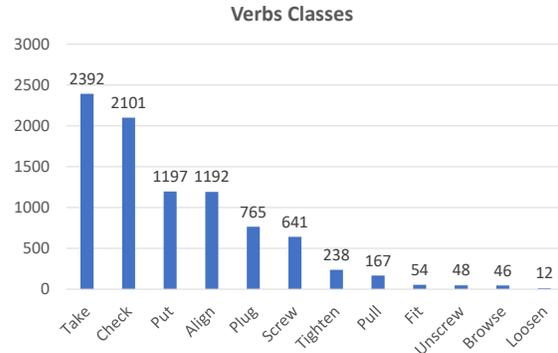


Figure 5. Long-tail distribution of verbs classes.

vironment. Objects of the same component class have been grouped and placed in a heap, and heaps have been placed randomly (see Figure 3). Other objects not related to the toy model were present in the scene (i.e., clutter background). We have considered two types of table: a light-colored table and a dark one. The dataset has been acquired by 20 different subjects in 2 countries (Italy and United Kingdom) between May 2019 and January 2020. Participants were from 8 different nationalities with ages between 18 and 55. Figure 4 reports some statistics about participants. We asked participants to sit and build the model of the motorbike. No other particular instruction was given to the participants, who were free to use all the objects placed in the table as well as the instruction booklet. Some examples of the captured data are reported in Figure 3.

Data was captured using an Intel RealSense SR300 device which has been mounted on the head of the participant with a headset. The headset was adjusted to control the point of view of the camera with respect to the different heights and postures of the participants, in order to have the hands located approximately in the middle of the scene to be acquired. Videos were recorded at a resolution of 1280x720 pixels and with a framerate of 12fps. Each video corresponds to a complete assembly of the toy model starting from the 49 pieces placed on the table. The average duration of the captured videos is 21.14min, with the longest one being 35.45min and the shortest one

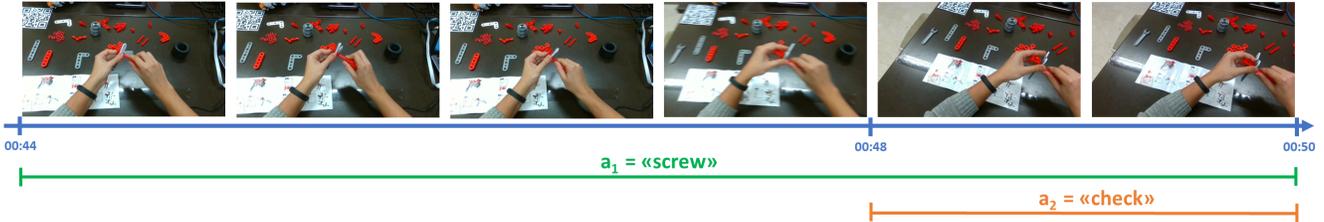


Figure 6. Example of two overlapping temporal annotations along with the associated verbs.

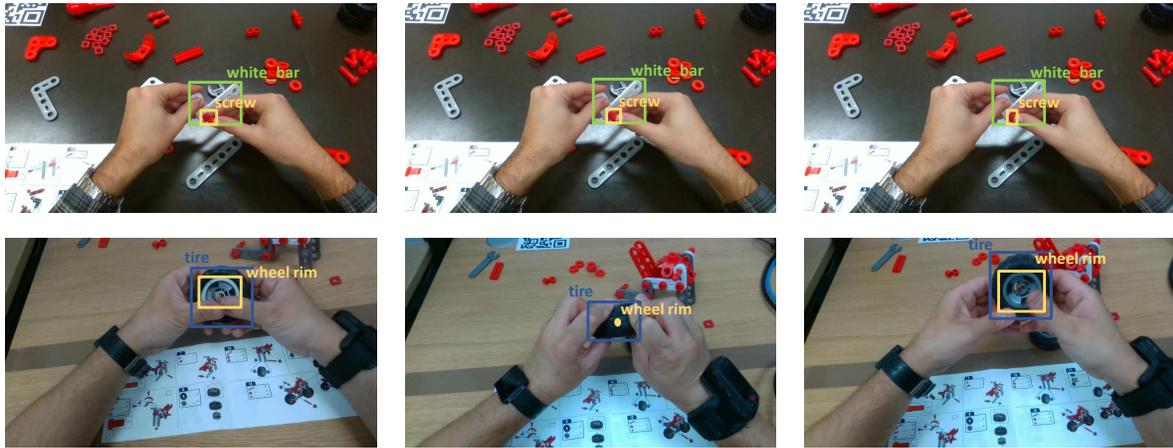


Figure 7. Example of bounding box annotations for *active* objects (first row) and occluded *active* objects (second row).

being 9.23min.

### 3.2. Data Annotation

We annotated the MECCANO dataset in two stages. In the first stage, we temporally annotated the occurrences of all human-object interactions indicating their start and end times, as well as a verb describing the interaction. In the second stage, we annotated the *active objects* with bounding boxes for each temporal segment.

**Stage 1: Temporal Annotations** We considered 12 different verbs which describe the actions performed by the participants: *take*, *put*, *check*, *browse*, *plug*, *pull*, *align*, *screw*, *unscrew*, *tighten*, *loosen* and *fit*. As shown in Figure 5, the distribution of verb classes of the labeled samples in our dataset follows a long-tail distribution, which suggests that the taxonomy captures the complexity of the considered scenario. Since a participant can perform multiple actions simultaneously, we allowed the annotated segments to overlap (see Figure 6). In particular, in the MECCANO dataset there are 1401 segments (15.82 %) which overlap with at least another segment. We consider the start time of a segment as the timestamp in which the hand touches an object, changing its state from *passive* to *active*. The only exception is for the verb *check*, in which case the user doesn't need to touch an object to perform an interaction.

In this case, we annotated the start time when it is obvious from the video sequence that the user is looking at the object (see Figure 6). With this procedure, we annotated 8857 video segments.

#### Stage 2: Active Object Bounding Box Annotations

We considered 20 object classes which include the 16 classes categorizing the 49 components, the two tools (*screwdriver* and *wrench*), the instructions booklet and a *partial\_model* class. The latter object class represents assembled components of the toy model which are not yet complete (e.g., a *screw* and a *bolt* fixed on a *bar* which have not yet been assembled with the rest of the model<sup>4</sup>). For each temporal segment, we annotated the *active* objects in frames sampled every 0.2 seconds. Each active object annotation consists in a  $(class, x, y, w, h)$  tuple, where *class* represents the class of the object and  $(x, y, w, h)$  defines a bounding box around the object. We annotated multiple objects when they were *active* simultaneously (see Figure 7 - first row). Moreover, if an active object is occluded, even just in a few frames, we annotated it with a  $(class, x, y)$  tuple, specifying the class of the object and its estimated 2D position. An example of occluded active object annotation is reported in the second row of Figure 7. With this procedure, we labeled a total of 64349 frames.

#### Action Annotations

Starting from the temporal anno-

<sup>4</sup>See the supplementary material for examples of partial model.

Split	#Videos	Duration (min)	%	#EHOIs Segments	Bounding Boxes	Country (U.K/Italy)	Table (Light/Dark)
Train	11	236.47	55%	5057	37386	6/5	6/5
Val	2	46.57	10%	977	6983	1/1	1/1
Test	7	134.93	35%	2824	19980	4/3	4/3

Table 2. Statistics of the three splits: Train, Validation and Test.

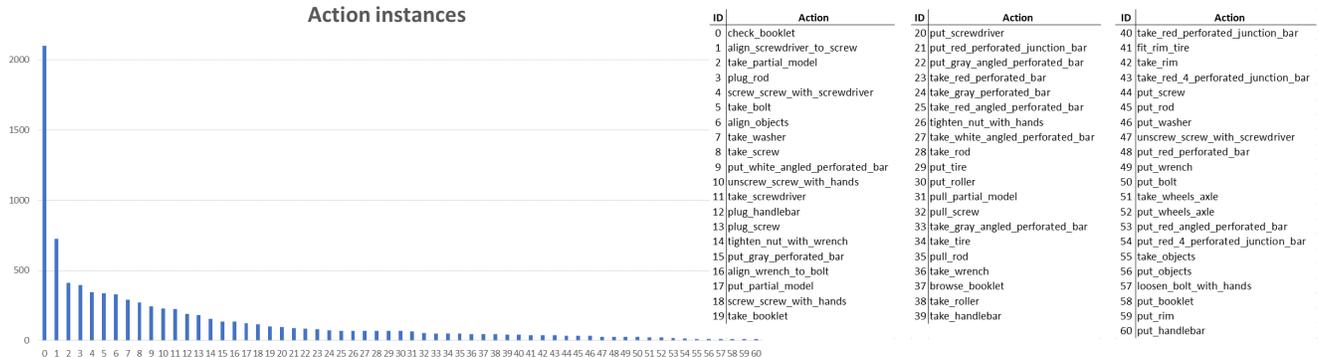


Figure 8. Distribution of action instances in the MECCANO dataset.

tations, we defined 61 action classes<sup>5</sup>. Each action is composed by a verb and one or more objects, for example “*align screwdriver to screw*” in which the verb is *align* and the objects are *screwdriver* and *screw*. Depending on the verb and objects involved in the interaction, each temporal segment has been associated to one of the 61 considered action classes. Figure 8 shows the list of the 61 action classes, which follow a long-tail distribution.

**EHOI Annotations** Let  $O = \{o_1, o_2, \dots, o_n\}$  and  $V = \{v_1, v_2, \dots, v_m\}$  be the sets of objects and verbs respectively. We define an Egocentric Human-Object Interaction  $e$  as:

$$e = (v_h, \{o_1, o_2, \dots, o_i\}) \quad (1)$$

where  $v_h \in V$  is the verb characterizing the interaction and  $(o_1, o_2, \dots, o_i) \subseteq O$  represent the active objects involved in the interaction. Given the previous definition, we considered all the observed combinations of verbs and objects to represent EHOIs performed by the participants during the acquisition (see examples in Figure 2-bottom). Each EHOI annotation is hence composed of a verb annotation and the *active* object bounding boxes. The MECCANO dataset is the first dataset of egocentric videos explicitly annotated for the EHOI detection task.

## 4. Benchmarks and Baseline Results

The MECCANO dataset is suitable to study a variety of tasks, considering the challenging industrial-like scenario in which it was acquired. In this paper, we consider four tasks for which we provide baseline results: 1) *Action Recognition*, 2) *Active Object Detection*, 3) *Active Object Recognition* and 4) *Egocentric Human-Object Interaction (EHOI)*

<sup>5</sup>See the supplementary material for details on action class selection.

**Detection.** While some of these tasks have been considered in previous works, none of them has been studied in industrial scenarios from the egocentric perspective. Moreover, it is worth noting that the EHOI Detection task has never been treated in previous works. We split the dataset into three subsets (*Training*, *Validation* and *Test*) designed to balance the different types of desks (light, dark) and countries in which the videos have been acquired (IT, U.K.). Table 2 reports some statistics about the three splits, such as the number of videos, the total duration (in seconds), the number of temporally annotated EHOIs and the number of bounding box annotations.

### 4.1. Action Recognition

**Task:** Action Recognition consists in determining the action performed by the camera wearer from an egocentric video segment. Specifically, let  $C_a = \{c_1, c_2, \dots, c_n\}$  be the set of action classes and let  $A_i = [t_{s_i}, t_{e_i}]$  be a video segment, where  $t_{s_i}$  and  $t_{e_i}$  are the start and the end times of the action respectively. The aim is to assign the correct action class  $c_i \in C_a$  to the segment  $A_i$ .

**Evaluation Measures:** We evaluate action recognition using Top-1 and Top-5 accuracy computed on the whole test set. As class-aware measures, we report class-mean precision, recall and  $F_1$ -score.

**Baselines:** We considered 2D CNNs as implemented in the PySlowFast library [21] (C2D), I3D [7] and SlowFast [24] networks, which are state-of-the-art methods for action recognition. In particular, for all baselines we used the PySlowFast implementation based on a ResNet-50 [33] backbone pre-trained on Kinetics [34]. See supplementary material for implementation details.

**Results:** Table 3 reports the results obtained by the baselines for the action recognition task. All baselines obtained

	Top-1 Accuracy	Top-5 Accuracy	Avg Class Precision	Avg Class Recall	Avg Class F <sub>1</sub> -score
C2D [21]	41.92	71.95	37.6	38.76	36.49
I3D [7]	42.51	72.35	40.04	40.42	38.88
SlowFast [24]	<b>42.85</b>	<b>72.47</b>	<b>42.11</b>	<b>41.48</b>	<b>41.05</b>

Table 3. Baseline results for the action recognition task.

Method	AP (IoU > 0.5)
Hand Object Detector [53]	11.17%
Hand Object Detector [53] (Avg dist.)	11.10%
Hand Object Detector [53] (All dist)	11.34%
Hand Object Detector [53] + Objs re-training	20.18%
Hand Object Detector [53] + Objs re-training (Avg dist.)	33.33%
Hand Object Detector [53] + Objs re-training (All dist.)	<b>38.14%</b>

Table 4. Baseline results for the *active* object detection task.

similar performance in terms of Top-1 and Top-5 accuracy with SlowFast networks achieving slightly better performance. Interestingly, performance gaps are more consistent when we consider precision, recall and  $F_1$  scores, which is particularly relevant given the long-tailed distribution of actions in the proposed dataset (see Figure 8). Note that, in our benchmark, SlowFast obtained the best results with a Top-1 accuracy of 47.82 and an  $F_1$ -score of 41.05. See supplementary material for qualitative results. In general, the results suggest that action recognition with the MECCANO dataset is challenging and offers a new scenario to compare action recognition algorithms.

## 4.2. Active Object Detection

**Task:** The aim of the Active Object Detection task is to detect all the *active* objects involved in EHOIs. Let  $O_{act} = \{o_1, o_2, \dots, o_n\}$  be the set of *active* objects in the image. The goal is to detect with a bounding box each *active* object  $o_i \in O_{act}$ .

**Evaluation Measures:** As evaluation measure, we use Average Precision (AP), which is used in standard object detection benchmarks. We set the IoU threshold equal to 0.5 in our experiments.

**Baseline:** We considered the Hand-Object Detector proposed in [53]. The model has been designed to detect hands and objects when they are in contact. This architecture is based on Faster-RCNN [52] and predicts a box around the visible human hands, as well as boxes around the objects the hands are in contact with and a link between them. We used the Hand-Object Detector [53] pretrained on EPIC-Kitchens [13], EGTEA [38] and CharadesEGO [54] as provided by authors [53]. The model has been trained to recognize hands and to detect the *active* objects regardless their class. Hence, it should generalize to others domains. With default parameters, the Hand-Object Detector can find at most two *active* objects in contact with hands. Since our dataset tends to contain more *active* objects in a single EHOI (up to 7), we consider two variants of this model by changing the threshold on the distance between

hands and detected objects. In the first variant, the threshold is set to the average distance between hands and *active* objects on the MECCANO dataset. We named this variant “*Avg distance*”. In the second variant, we removed the thresholding operation and considered all detected objects as *active* objects. We named this variant “*All objects*”. We further adapted the Hand-Object Detector [53] re-training the Faster-RCNN component to detect all *active* objects of the MECCANO dataset. See supplementary material for implementation details.

**Results:** Table 4 shows the results obtained by the *active* object detection task baselines. The results highlight that the Hand-Object Detector [53] is not able to generalize to a domain different than the one on which it was trained. All the three variants of the Hand-Object Detector using the original object detector obtained an AP approximately equal to 11% (first three rows of Table 4). Re-training the object detector on the MECCANO dataset allowed to improve performance by significant margins. In particular, using the standard distance threshold value, we obtained an AP of 20.18%. If we consider the average distance as the threshold to discriminate *active* and *passive* objects, we obtain an AP of 33.33%. Removing the distance threshold (last row of Table 4), allows to outperform all the previous results obtaining an AP equal to 38.14%. This suggests that adapting the general object detector to the challenging domain of the proposed dataset is key to performance. Indeed, training the object detector to detect only *active* objects in the scene already allows to obtain reasonable results, while there still space for improvement.

## 4.3. Active Object Recognition

**Task:** The task consists in detecting and recognizing the *active* objects involved in EHOIs considering the 20 object classes of the MECCANO dataset. Formally, let  $O_{act} = \{o_1, o_2, \dots, o_n\}$  be the set of *active* objects in the image and let  $C_o = \{c_1, c_2, \dots, c_m\}$  be the set of object classes. The task consists in detecting objects  $o_i \in O_{act}$  and assigning them the correct class label  $c \in C_o$ .

**Evaluation Measures:** We use mAP [19] with threshold on IoU equal to 0.5 for the evaluations.

**Baseline:** As a baseline, we used a standard Faster-RCNN [52] object detector. For each image the object detector predicts  $(x, y, w, h, class)$  tuples which represent the object bounding boxes and the associated classes. See supplementary material for implementation details.

**Results:** Table 7 reports the results obtained with the base-

ID	Class	AP (per class)
0	instruction booklet	46.18%
1	gray_angled_perforated_bar	09.79%
2	partial_model	36.40%
3	white_angled_perforated_bar	30.48%
4	wrench	10.77%
5	screwdriver	60.50%
6	gray_perforated_bar	30.83%
7	wheels_axle	10.86%
8	red_angled_perforated_bar	07.57%
9	red_perforated_bar	22.74%
10	rod	15.98%
11	handlebar	32.67%
12	screw	38.96%
13	tire	58.91%
14	rim	50.35%
15	washer	30.92%
16	red_perforated_junction_bar	19.80%
17	red_4_perforated_junction_bar	40.82%
18	bolt	23.44%
19	roller	16.02%

mAP | 30.39%

Table 5. Baseline results for the *active* object recognition task.

line in the *Active* Object Recognition task. We report the AP values for each class considering all the videos belonging to the test set of the MECCANO dataset. The last column shows the average of the AP values for each class and the last row reports the mAP value for the test set. The mAP was computed as the average of the mAP values obtained in each test video. AP values in the last column show that large objects are easier to recognize (e.g. *instruction booklet*: 46.48%; *screwdriver*: 60.50%; *tire*: 58.91%; *rim*: 50.35%). Performance suggests that the proposed dataset is challenging due to the presence of small objects. We leave the investigation of more specific approaches to active object detection to future studies.

#### 4.4. EHOI Detection

**Task:** The goal is to determine egocentric human-object interactions (EHOI) in each image. Given the definition of EHOIs as  $\langle \text{verb}, \text{objects} \rangle$  pairs (see Equation 1), methods should detect and recognize all the *active* objects in the scene, as well as the verb describing the action performed by the human.

**Evaluation Measures:** Following [32, 29], we use the “*role AP*” as an evaluation measure. Formally, a detected EHOI is considered as a true positive if 1) the predicted object bounding box has a IoU of 0.5 or higher with respect to a ground truth annotation and 2) the predicted verb matches with the ground truth. Note that only the *active* object bounding box location (not the correct class) is considered in this measure. Moreover, we used different values of IoU (e.g., 0.5, 0.3 and 0.1) to compute the “*role AP*”.

**Baseline:** We adopted three baselines for the EHOI detec-

Model	mAP role		
	IoU $\geq$ 0.5	IoU $\geq$ 0.3	IoU $\geq$ 0.1
InteractNet [29]	04.92%	05.30%	05.72%
InteractNet [29] + Context	08.45%	09.01%	09.45%
SlowFast [24] + Faster-RCNN [52]	<b>25.93%</b>	<b>28.04%</b>	<b>29.65%</b>

Table 6. Baseline results for the EHOI detection task.



Figure 9. Qualitative results for the EHOI detection task.

tion task. The first one is based on InteractNet [29], which is composed by three branches: 1) the “human-branch” to detect the humans in the scene, 2) the “object-branch” to detect the objects and 3) the “interaction-branch” which predicts the verb of the interaction focusing on the humans and objects appearance. The second one is an extension of InteractNet which also uses context features derived from the whole input frame to help the “interaction-branch” in verb prediction. The last baseline is based on the combination of a SlowFast network [21] trained to predict the verb of the EHOI considering the spatial and temporal dimensions, and Faster-RCNN [52] which detects and recognizes all *active* objects in the frame. See supplementary material for implementation details.

**Results:** Table 6 reports the results obtained by the baselines on the test set for the EHOI detection task. The InteractNet method obtains low performance on this task with a mAP role of 4.92%. Its extension with context features, slightly improves the performance with a mAP role of 8.45%, whereas SlowFast network with Faster-RCNN achieved best results with a mAP equal to 25.93%. The results highlight that current state-of-the-art approaches developed for the analysis of still images in third person scenarios are unable to detect EHOIs in the proposed dataset, which is likely due to the presence of multiple tiny objects involved simultaneously in the EHOI and to the actions performed. On the contrary, adding the ability to process video clips with SlowFast allows for significant performance boosts. Figure 9 shows qualitative results obtained with the SlowFast+Faster-RCNN baseline. Note that in the second example the method correctly predicted all the objects involved simultaneously in the EHOI. Despite promising performance of the suggested baseline, the proposed EHOI detection task needs more investigation due to the challenging nature of the considered industrial-like domain.

## 5. Conclusion

We proposed MECCANO, the first dataset to study egocentric human-object interactions (EHOIs) in an industrial-like scenario. We publicly release the dataset with both temporal (action segments) and spatial (active object bounding boxes) annotations considering a taxonomy of 12 verbs, 20 nouns and 61 unique actions. In addition, we defined the Egocentric Human-Object Interaction (EHOI) detection task and performed baseline experiments to show the potential of the proposed dataset on four challenging tasks: action recognition, *active* object detection, *active* object recognition and EHOI detection. Future works will explore approaches for improved performance on this challenging data.

## Acknowledgments

This research has been supported by MIUR PON PON R&I 2014-2020 - Dottorati innovativi con caratterizzazione industriale, by MIUR AIM - Attrazione e Mobilita Internazionale Linea 1 - AIM1893589 - CUP: E64118002540007, and by MISE - PON I&C 2014-2020 - Progetto ENIGMA - Prog n. F/190050/02/X44 – CUP: B61B19000520008.

## SUPPLEMENTARY MATERIAL

This document is intended for the convenience of the reader and reports additional information about the proposed dataset, the annotation stage, as well as implementation details related to the performed experiments. This supplementary material is related to the following submission:

- F. Ragusa, A. Furnari, S. Livatino, G. M. Farinella. The MECCANO Dataset: Understanding Human-Object Interactions from Egocentric Videos in an Industrial-like Domain, submitted to IEEE Winter Conference on Applications of Computer Vision (WACV), 2021.

The remainder of this document is organized as follows. Section 6 reports additional details about data collection and annotation. Section 7 provides implementation details of the compared methods. Section 8 reports additional qualitative results.

## 6. Additional details on the MECCANO Dataset

### 6.1. Component classes and grouping

The toy motorbike used for our data collection is composed of 49 components belonging to 19 classes (Figure 1), plus two tools. In our settings, we have grouped two types

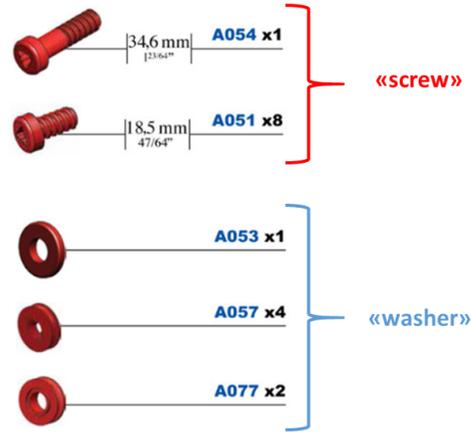


Figure 10. Grouped pieces belonging to *screw* and *washer* classes.

of components which are similar in their appearance and have similar roles in the assembly process. Figure 10 illustrates the two groups. Specifically, we grouped A054 and A051 under the “screw” class. These two types of components only differ in their lengths. We also grouped A053, A057 and A077 under the “washers” class. Note that these components only differ in the radius of their holes and in their thickness.

As a results, we have 20 object classes in total: 16 classes are related to the 49 motorbike components, whereas the others are associated to the two tools, to the instruction booklet and to a partial model class, which indicates a set of components assembled together to form a part of the model (see Figure 11 ).

### 6.2. Data Annotation

**Verb Classes and Temporal Annotations** We considered 12 verb classes which describe all the observed actions performed by the participants during the acquisitions. Figure 12 reports the percentage of the temporally annotated instances belonging to the 12 verb classes. The considered verb classes are: *take*, *put*, *check*, *browse*, *plug*, *pull*, *align*, *screw*, *unscrew*, *tighten*, *loosen* and *fit*. We used the ELAN Annotation tool [45] to annotate a temporal segment around each instance of an action. Each segment has been associated to the verb which best described the contained action.

**Active Object Bounding Box Annotations** For each annotated video segment, we sampled frames every 0.2 seconds. Each of these frames has been annotated to mark the presence of all *active* objects with bounding boxes and related component class label. To this aim, we used VGG Image Annotator (VIA) [18] with a customized project which allowed annotators to select component classes from a dedicated panel showing the thumbnails of each of the 20 object classes to facilitate and speed up the selection of the correct object class. Figure 13 reports an example of the

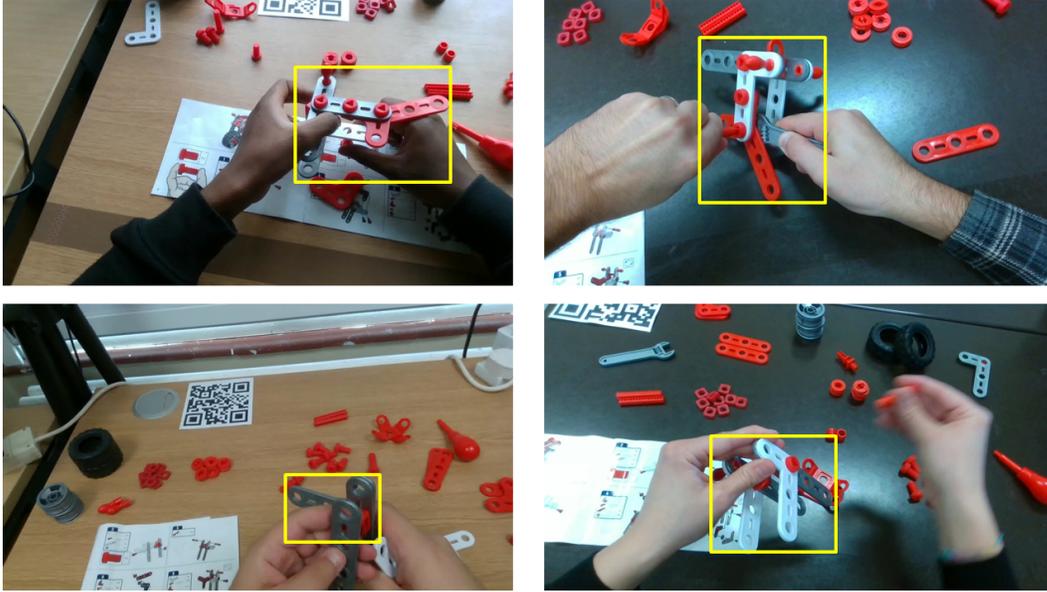


Figure 11. Examples of objects belonging to the partial model class.

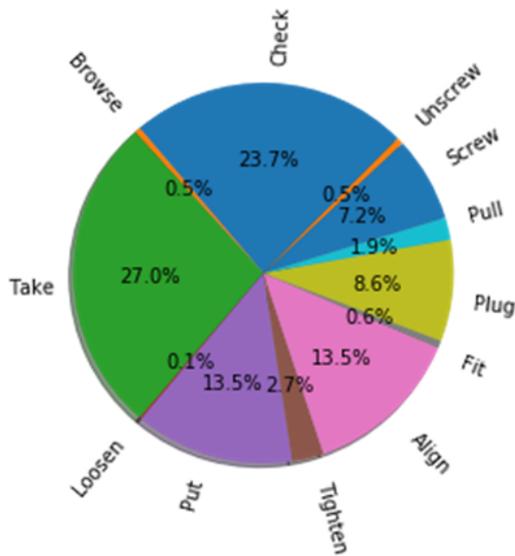


Figure 12. Fractions of instances of each verb in the MECCANO dataset.

customized VIA interface. Moreover, to support annotators and reduce ambiguities, we prepared a document containing a set of fundamental rules for the annotations of *active* objects, where we reported the main definitions (e.g., active object, occluded active object, partial\_model) along with visual examples. Figure 14 reports an example of such instructions.

**Action Annotation** In the MECCANO dataset, an action can be seen as a combination of a verb and a set of nouns (e.g., “take wrench”). We analyzed the combinations

of our 12 verb classes and 20 object classes to find a compact, yet descriptive set of actions classes. The action class selection process has been performed in two stages. In the first stage, we obtained the distributions of the number of active objects generally occurring with each of the 12 verbs. The distributions are shown in Figure 15. For example, the dataset contains 120 instances of “browse” (second row - first column), which systematically involves one single object. Similarly, most of the instance of “take” appear with 1 object, while few instances have 2 – 3 objects.

In the second stage, we selected a subset of actions from all combinations of verbs and nouns. Figure 16 reports all the action classes obtained from the 12 verbs classes of the MECCANO dataset as discussed in the following. Let  $O = \{o_1, o_2, \dots, o_n\}$  and  $V = \{v_1, v_2, \dots, v_m\}$  be the set of the objects and verb classes respectively. For each verb  $v \in V$ , we considered all the object classes  $o \in O$  involved in one or more temporal segments labeled with verb  $v$ . We considered the following rules:

- **Take and put:** We observed that all the objects  $o \in O$  occurring with  $v = take$  are taken by participants while they build the motorbike. Hence, we first defined 20 action classes as  $(v, o)$  pairs (one for each of the available objects). Since subjects can take more than one object at a time, we added an additional “take objects” action class when two or more objects are taken simultaneously. The same behavior has been observed for the verb  $v = put$ . Hence, we similarly defined 21 action classes related to this verb.
- **Check and browse:** We observed that verbs  $v = check$  and  $v = browse$  always involve only the object

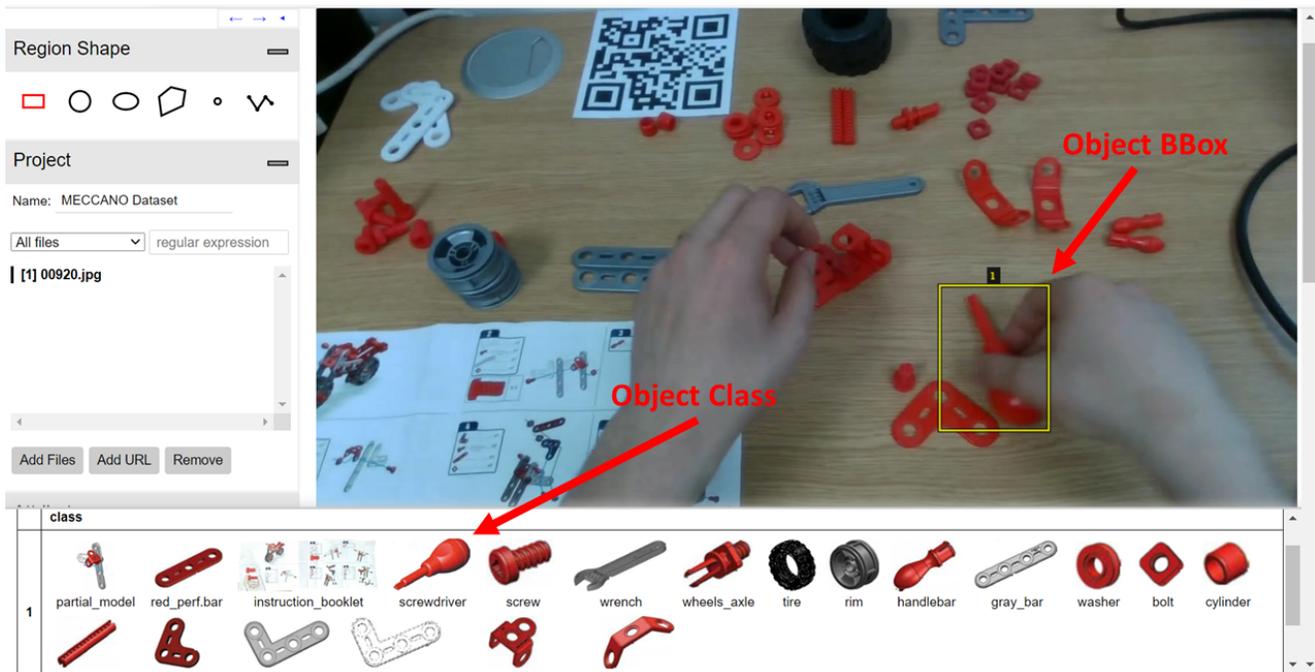


Figure 13. Customized VIA project to support the labeling of active objects. Annotators were presented with a panel which allowed them to identify object classes through their thumbnails.

#### Definitions:

**Active Object:** the object which is involved in the action. Without this object, the action loses its meaning.

**Example:** Take wrench

The action **take wrench** without the object **wrench** loses its meaning. In this case, you should annotate the object **wrench** with a bounding box around it.

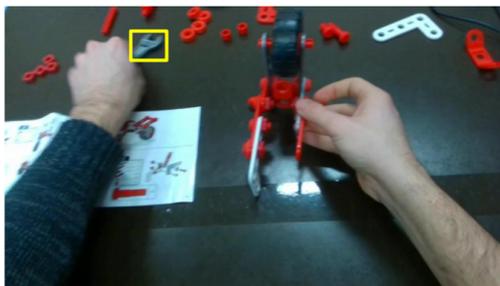


Figure 14. Active object definition given to the labelers for the active object bounding box annotation stage.

$o = instruction\ booklet$ . Hence, we defined the two action classes *check instruction booklet* and *browse instruction booklet*.

- **Fit:** When the verb is  $v = fit$ , there are systematically two objects involved simultaneously (i.e.,  $o = rim$

and  $o = tire$ ). Hence, we defined the action class *fit rim and tire*.

- **Loosen:** We observed that participants tend to loosen bolts always with the hands. We hence defined the action class *loosen bolt with hands*.
- **Align:** We observed that participants tend to align the screwdriver tool with the screw before starting to screw, as well as the wrench tool with the bolt before tightening it. Participants also tended to align objects to be assembled to each other. From these observations, we defined three action classes related to the verb  $v = align$ : *align screwdriver to screw*, *align wrench to bolt* and *align objects*.
- **Plug:** We found three main uses of verb  $v = plug$  related to the objects  $o = screw$ ,  $o = rod$  and  $o = handlebar$ . Hence, we defined three action classes: *plug screw*, *plug rod* and *plug handlebar*.
- **Pull:** Similar observations apply to verb  $v = pull$ . Hence we defined three action classes involving “pull”: *pull screw*, *pull rod* and *pull partial model*.
- **Screw and unscrew:** The main object involved in actions characterized by the verbs  $v = screw$  and  $v = unscrew$  is  $o = screw$ . Additionally, the screw

or unscrew action can be performed with a screwdriver or with hands. Hence, we defined four action classes *screw screw with screwdriver*, *screw screw with hands*, *unscrew screw with screwdriver* and *unscrew screw with hands*.

- **Tighten:** Similar observation holds for the verb  $v = \textit{tighten}$ , the object  $o = \textit{bolt}$  and the tool  $o = \textit{wrench}$ . We hence defined the following two action classes: *tighten bolt with wrench* and *tighten bolt with hands*.

In total, we obtained 61 action classes composing the MECCANO dataset.

## 7. Baseline Implementation Details

### 7.1. Action Recognition

The goal of action recognition is to classify each action segment into one of the 61 action classes of the MECCANO dataset. The SlowFast, C2D and I3D baselines considered in this paper all require fixed-length clips at training time. Hence, we temporally downsample or upsample uniformly each video shot before passing it to the input layer of the network. The average number of frames in a video clip in the MECCANO dataset is 26.19. For SlowFast network, we set  $\alpha = 4$  and  $\beta = \frac{1}{8}$ . We set the batch-size to 12 for C2D and I3D, we used a batch-size of 20 for SlowFast. We trained C2D, I3D and SlowFast networks on 2 NVIDIA V100 GPUs for 80, 70 and 40 epochs with learning rates of 0.01, 0.1 and 0.0001 respectively. These settings allowed all baselines to converge.

### 7.2. Active Object Detection

We trained Faster-RCNN on the training and validation sets using the provided *active* object labels. We set the learning rate to 0.005 and trained Faster-RCNN with a ResNet-101 backbone and Feature Pyramid Network for 100K iterations on 2 NVIDIA V100 GPUs. We used the Detectron2 implementation [65]. The model is trained to recognize objects along with their classes. However, for the active object detection task, we ignore output class names and only consider a single “active object” class.

### 7.3. Active Object Recognition

We used the same model adopted for the Active Object Detection task, retaining also object classes at test time.

### 7.4. EHOI Detection

For the “SlowFast + Faster-RCNN” baseline, we trained SlowFast network to recognize the 12 verb classes of the MECCANO dataset using the same settings as the ones considered for the action recognition task. We trained the network for 40 epochs and obtained a verb recognition Top-1

accuracy of 58.04% on the Test set. For the object detector component, we used the same model trained for the active object recognition task.

For the “human-branch” of the “InteractNet” model, we used the Hand-Object Detector [53] to detect hands in the scene. The object detector trained for active object recognition has been used for the “object-branch”. The MLPs used to predict the verb class from the appearance of hands and active objects are composed by an input linear layer (e.g., 1024-d for the hands MLP and 784-d for the objects one), a ReLU activation function and an output linear layer (e.g., 12-d for both MLPs). We fused by late fusion the output probability distributions of verbs obtained from the two MLPs (hands and objects) to predict the final verb of the EHOI. We jointly trained the MLPs for 50K iterations on an Nvidia V100 GPU, using a batch size of 28 and a learning rate of 0.0001.

In “InteractNet + Context”, we added a third MLP which predicts the verb class based on context features. The context MLP has the same architecture of the others MLPs (hands and objects) except the input linear layer which is 640-d. In this case, we jointly trained the three MLPs (hands, objects and context) for 50K iterations on a TitanX GPU with a batch size equal to 18 and the learning rate equal to 0.0001. The outputs of the three MLPs are hence fused by late fusion.

## 8. Additional Results

Figure 17 shows some qualitative results of the SlowFast baseline. Note that, in the second and third example, the method predicts correctly only the verb or the object.

Table 7 reports the results obtained with the baseline in the *Active* Object Recognition task. We report the AP values for each class considering all the videos belonging to the test set of the MECCANO dataset. The last column shows the average of the AP values for each class and the last row reports the mAP values for each test video. Figure 18 reports some qualitative results for this task. In particular, in the first row, we report the correct *active* object predictions, while in the second row we report two examples of wrong predictions. In the wrong predictions, the right *active* object is recognized but other *passive* objects are wrongly detected and recognized as *active* (e.g., instruction booklet in the example bottom-left or the red bars in the example bottom-right of Figure 18).

## References

- [1] Epson moverio bt 300. <https://www.epson.eu/products/see-through-mobile-viewer/moverio-bt-300>.
- [2] Microsoft hololens 2. <https://www.microsoft.com/en-us/hololens>.

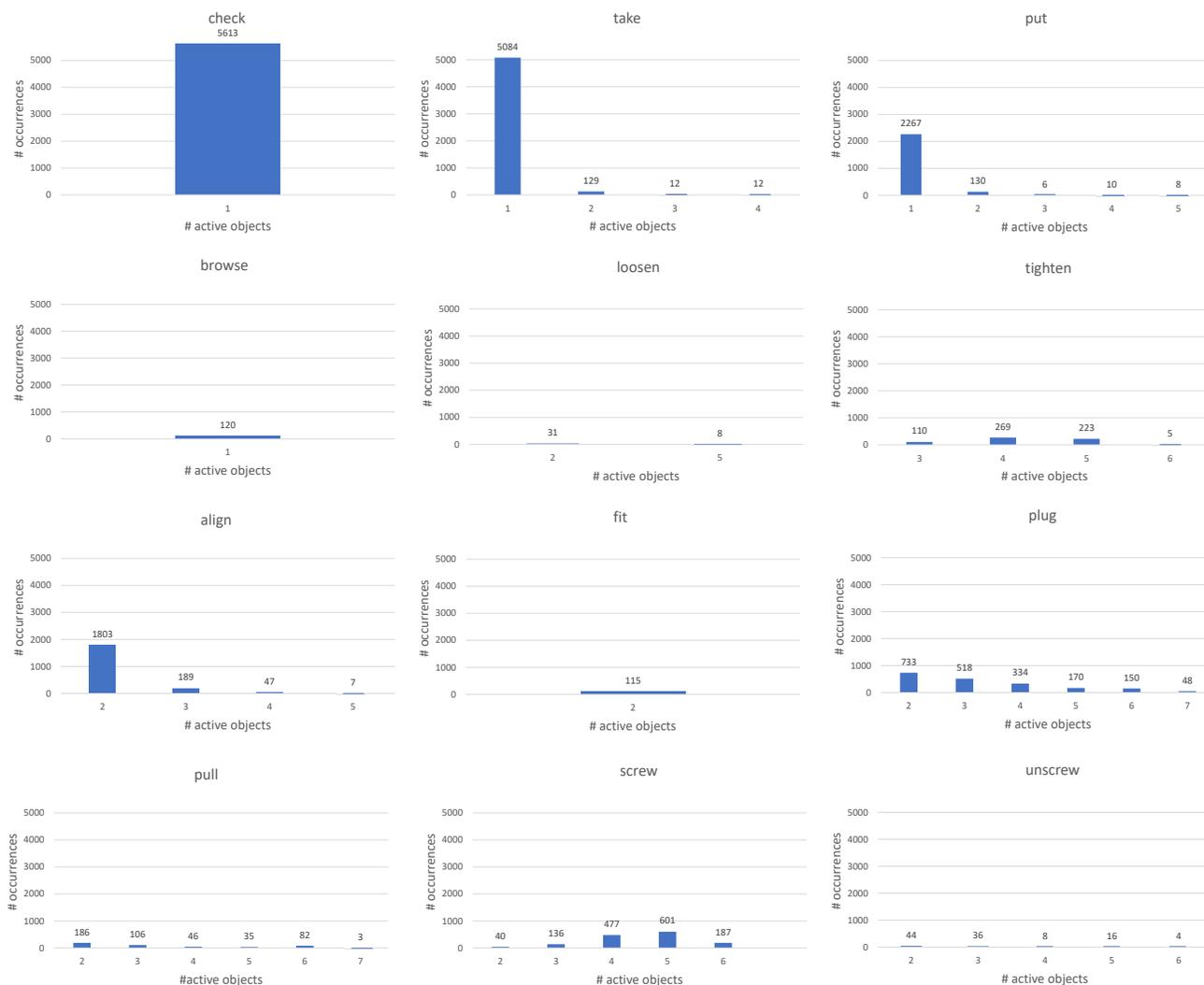


Figure 15. Number of objects and occurrences of *active* objects related to each verb.

- [3] Vuzix blade. <https://www.vuzix.com/products/blade-smart-glasses>.
- [4] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *ICCV*, pages 1949–1957, 2015.
- [5] Minjie Cai, Kris M. Kitani, and Yoichi Sato. Understanding hand-object manipulation with grasp types and object attributes. In *Robotics: Science and Systems*, 2016.
- [6] J. Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *ArXiv*, abs/1907.06987, 2019.
- [7] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *CVPR*, pages 4724–4733, 2017.
- [8] Y. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. Hico: A benchmark for recognizing human-object interactions in images. In *ICCV*, pages 1017–1025, 2015.
- [9] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. *WACV*, pages 381–389, 2018.
- [10] Sara Colombo, Yihyun Lim, and Federico Casalegno. Deep vision shield: Assessing the use of hmd and wearable sensors in a smart safety device. In *ACM PETRA*, 2019.
- [11] Rita Cucchiara and Alberto Del Bimbo. Visions for augmented cultural heritage experience. *IEEE MultiMedia*, 21(1):74–82, 2014.
- [12] Naveet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, page 428–441, 2006.
- [13] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

ID	Class\Video	0008	0009	0010	0011	0012	0019	0020	AP (per class)
0	instruction booklet	62.00%	38.78%	42.97%	63.75%	29.84%	38.25%	47.65%	46.18%
1	gray_angled_perforated_bar	9.55%	18.81%	14.72%	2.17%	16.42%	0%	6.89%	9.79%
2	partial_model	35.68%	31.74%	35.82%	42.55%	32.16%	33.02%	43.80%	36.40%
3	white_angled_perforated_bar	43.70%	39.86%	9.90%	45.32%	24.94%	16.35%	33.31%	30.48%
4	wrench	//	//	//	11.11%	//	10.43%	//	10.77%
5	screwdriver	61.82%	57.68%	68.57%	54.21%	57.14%	62.68%	61.37%	60.50%
6	gray_perforated_bar	19.36%	40.26%	30.89%	53.06%	29.68%	26.82%	15.76%	30.83%
7	wheels_axle	11.37%	18.34%	04.63%	1.79%	31.61%	03.91%	04.35%	10.86%
8	red_angled_perforated_bar	18.65%	01.57%	4.81%	00.09%	12.27%	05.98%	09.64%	07.57%
9	red_perforated_bar	23.35%	26.69%	34.72%	24.58%	20.70%	11.21%	17.91%	22.74%
10	rod	14.90%	07.40%	22.41%	19.73%	15.57%	17.84%	14.04%	15.98%
11	handlebar	44.39%	36.31%	28.79%	26.92%	12.50%	27.27%	52.48%	32.67%
12	screw	48.64%	42.87%	40.00%	16.96%	44.99%	43.88%	35.35%	38.96%
13	tire	45.93%	71.68%	63.09%	89.01%	37.83%	39.69%	65.15%	58.91%
14	rim	45.10%	35.71%	42.57%	59.26%	22.28%	90.00%	57.54%	50.35%
15	washer	31.52%	39.39%	19.00%	19.57%	53.43%	44.45%	09.06%	30.92%
16	red_perforated_junction_bar	19.28%	13.51%	07.55%	30.74%	28.63%	22.02%	16.89%	19.80%
17	red_4_perforated_junction_bar	24.20%	43.50%	39.11%	85.71%	44.23%	28.37%	20.62%	40.82%
18	bolt	33.14%	33.61%	11.29%	17.16%	28.46%	21.31%	19.12%	23.44%
19	roller	09.93%	40.50%	28.15%	5.76%	0.23%	18.20%	09.36%	16.02%
<b>mAP (per video)</b>		31.71%	33.59%	28.89%	33.47%	28.57%	28.08%	28.44%	<b>30.39%</b>

Table 7. Baseline results for the *active* object recognition task. We report the AP values for each class which are the averages of the AP values for each class of the Test videos. In the last column, we report the mAP per class, which is the average mAP of the Test videos.

- [14] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE TPAMI*, 2020.
- [15] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020.
- [16] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*, 2014.
- [17] Fernando de la Torre, Jessica K. Hodgins, Javier Montano, and Sergio Valcarcel. Detailed human data acquisition of kitchen activities: the cmu-multimodal activity database (cmu-mmact). In *CHI 2009 Workshop. Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009.
- [18] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, New York, NY, USA, 2019. ACM.
- [19] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vision*, 111(1):98–136, Jan. 2015.
- [20] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.
- [21] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. <https://github.com/facebookresearch/slowfast>, 2020.
- [22] K. Fang, T. Wu, D. Yang, S. Savarese, and J. J. Lim. Demo2vec: Reasoning object affordances from online videos. In *CVPR*, pages 2139–2147, 2018.
- [23] G. M. Farinella, G. Signorello, S. Battiato, A. Furnari, F. Ragusa, R. Leonardi, E. Ragusa, E. Scuderi, A. Lopes, L. Santo, and M. Samarotto. VEDI: Vision exploitation for data interpretation. In *ICIAP*, 2019.
- [24] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2018.
- [25] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. In *NeurIPS, NIPS'16*, page 3476–3484, Red Hook, NY, USA, 2016. Curran Associates Inc.
- [26] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.
- [27] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.
- [28] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [29] Georgia Gkioxari, Ross B. Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. *CVPR*, pages 8359–8367, 2018.
- [30] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Freund, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thureau, I. Bax, and R. Memisevic. The “something something” video database for learning and evaluating visual common sense. In *ICCV*, pages 5843–5851, 2017.
- [31] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE TPAMI*, 31(10), 2009.

- [32] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *ArXiv*, abs/1505.04474, 2015.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [34] W. Kay, J. Carreira, K. Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017.
- [35] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.
- [36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NeurIPS*, pages 1097–1105. 2012.
- [37] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*. IEEE Computer Society, 2008.
- [38] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018.
- [39] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. PPDm: Parallel point detection and matching for real-time human-object interaction detection. In *CVPR*, 2020.
- [40] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7082–7092, 2019.
- [41] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. cite arxiv:1405.0312.
- [42] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. In *CVPR*, 2016.
- [43] T. Nagarajan, C. Feichtenhofer, and K. Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, pages 8687–8696, 2019.
- [44] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. *ArXiv*, abs/2001.04583, 2020.
- [45] The Language Archive Nijmegen: Max Planck Institute for Psycholinguistics. Elan (version 5.9) [computer software]. 2020.
- [46] A. Ortis, G. Farinella, V. D’Amico, Luca Addesso, Giovanni Torrisi, and S. Battiato. Organizing egocentric videos of daily living activities. *Pattern Recognition*, 72, 2017.
- [47] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012.
- [48] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. *ArXiv*, abs/1808.07962, 2018.
- [49] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. pages 5534–5542, 10 2017.
- [50] F. Ragusa, A. Furnari, S. Battiato, G. Signorello, and G. M. Farinella. EGO-CH: Dataset and fundamental tasks for visitors behavioral understanding using egocentric vision. *Pattern Recognition Letters*, 2020.
- [51] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.
- [52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks.
- [53] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. 2020.
- [54] Gunnar A. Sigurdsson, Abhinav Gupta, C. Schmid, Ali Farhadi, and Alahari Karteek. Actor and observer: Joint modeling of first and third-person videos. *CVPR*, pages 7396–7404, 2018.
- [55] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.
- [56] Bilge Soran, Ali Farhadi, and Linda Shapiro. Generating notifications for missing actions: Don’t forget to turn the lights off! pages 4669–4677, 12 2015.
- [57] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *CVPR*, June 2019.
- [58] Y. Tang, Y. Tian, J. Lu, J. Feng, and J. Zhou. Action recognition in rgb-d egocentric videos. In *ICIP*, pages 3410–3414, 2017.
- [59] Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, page 140–153, 2010.
- [60] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.
- [61] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018.
- [62] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE TPAMI*, 40(6):1510–1517, 2018.
- [63] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. volume 9912, 10 2016.
- [64] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *CVPR*, June 2020.
- [65] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [66] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *CoRR*, abs/1712.04851, 2017.

- [67] B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE TPAMI*, 34(9):1691–1703, 2012.
- [68] Bolei Zhou, Alex Andonian, and Antonio Torralba. Temporal relational reasoning in videos. *ArXiv*, abs/1711.08496, 2018.
- [69] P. Zhou and M. Chi. Relation parsing neural network for human-object interaction detection. In *ICCV*, pages 843–851, 2019.

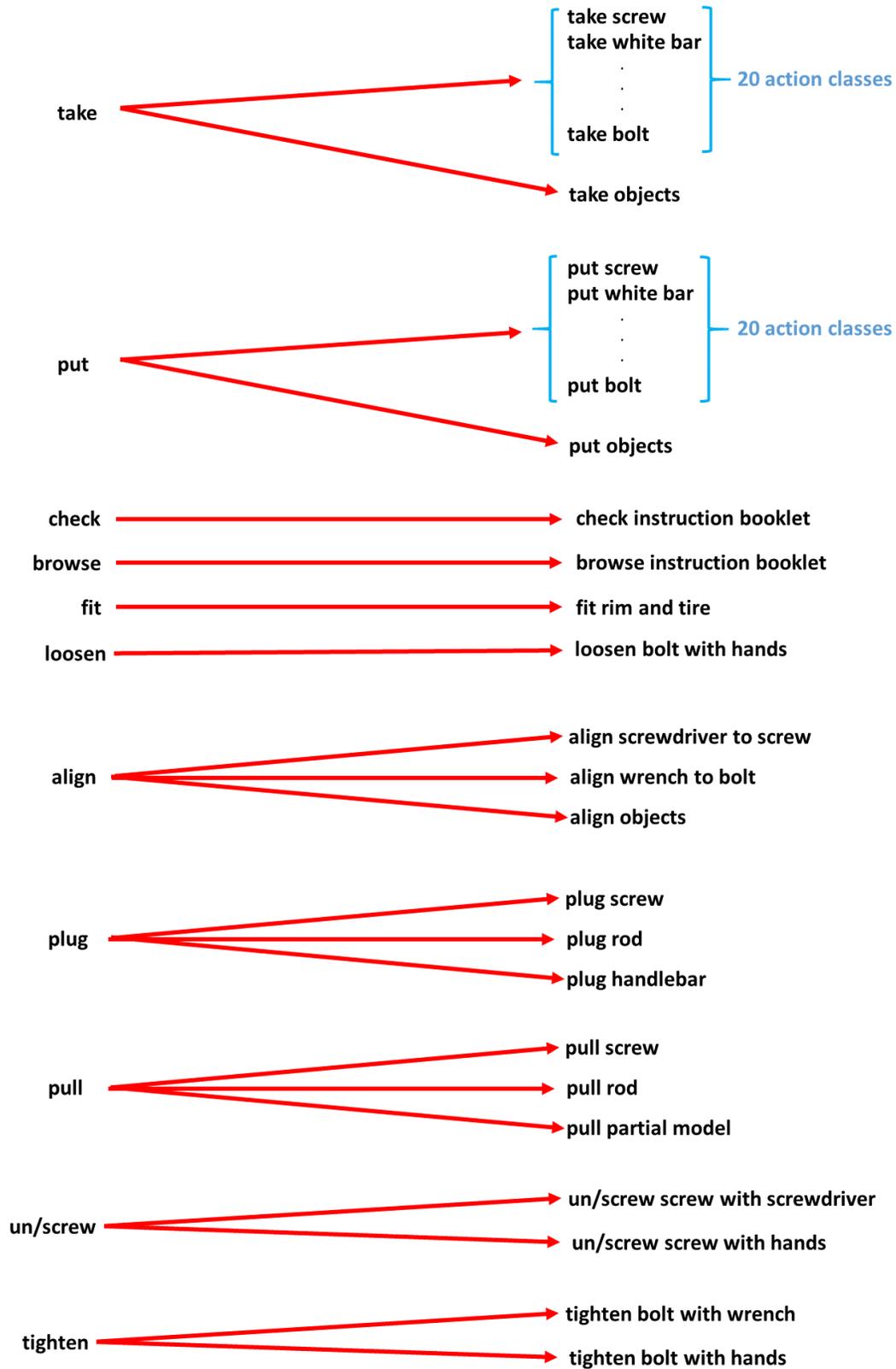


Figure 16. 61 action classes definition from the 12 verb classes and the analysis performed observing the participant behavior.

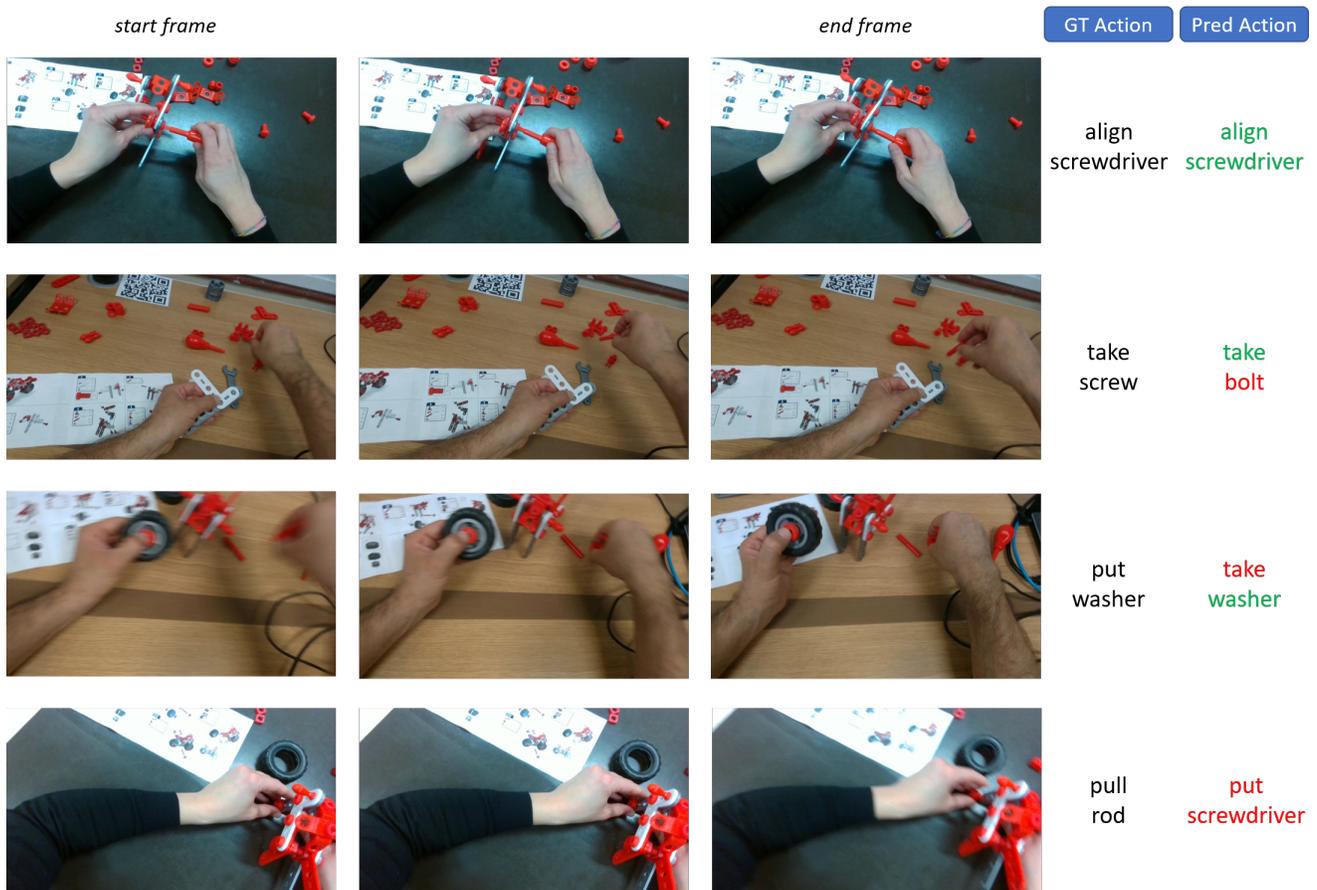


Figure 17. Qualitative results for the action recognition task. Correct predictions are in green while wrong predictions are in red.

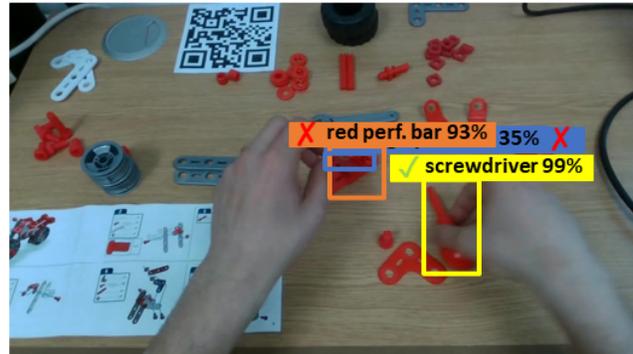
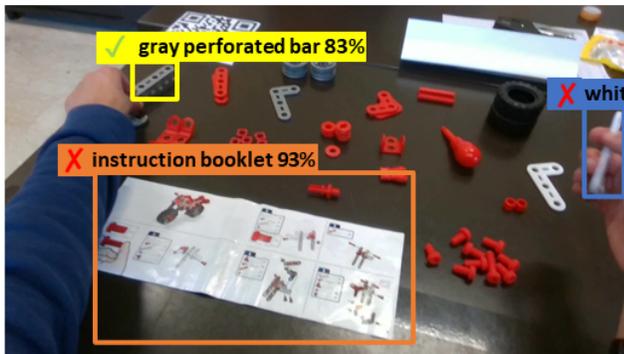
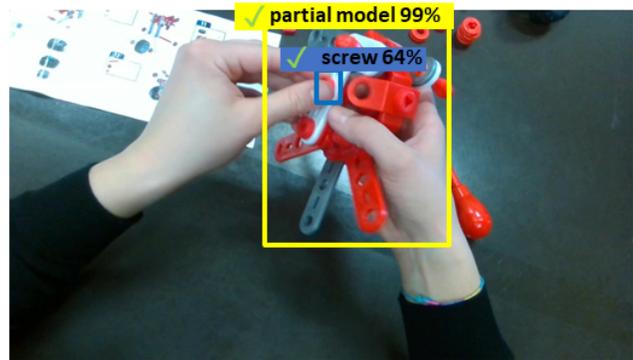
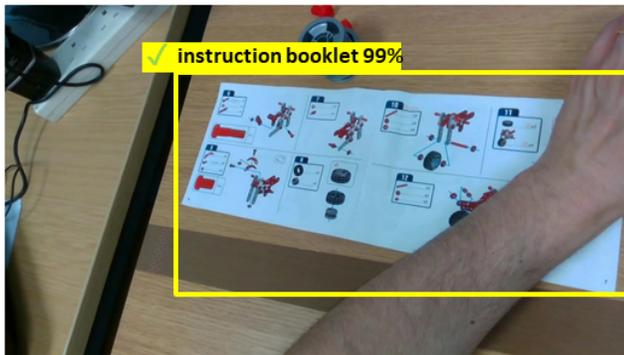


Figure 18. Qualitative results for the *active* object recognition task.