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

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

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

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

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

1. Introduction

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. In IEEE Winter Conference on Applications of Computer Vision (WACV), 2021.

The reader is referred to the manuscript and to our web page https://iplab.dmi.unict.it/MECCANO/ to download the dataset and for further information.

The remainder of this document is organized as follows. Section 2 reports additional details about data collection and annotation. Section 3 provides implementation details of the compared methods. Section 4 reports additional qualitative results.

2. Additional details on the MECCANO Dataset

2.1. 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 of components which are similar in their appearance and have similar roles in the assembly process. Figure 2 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 3).

2.2. 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 4 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 [2] 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) [1] 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

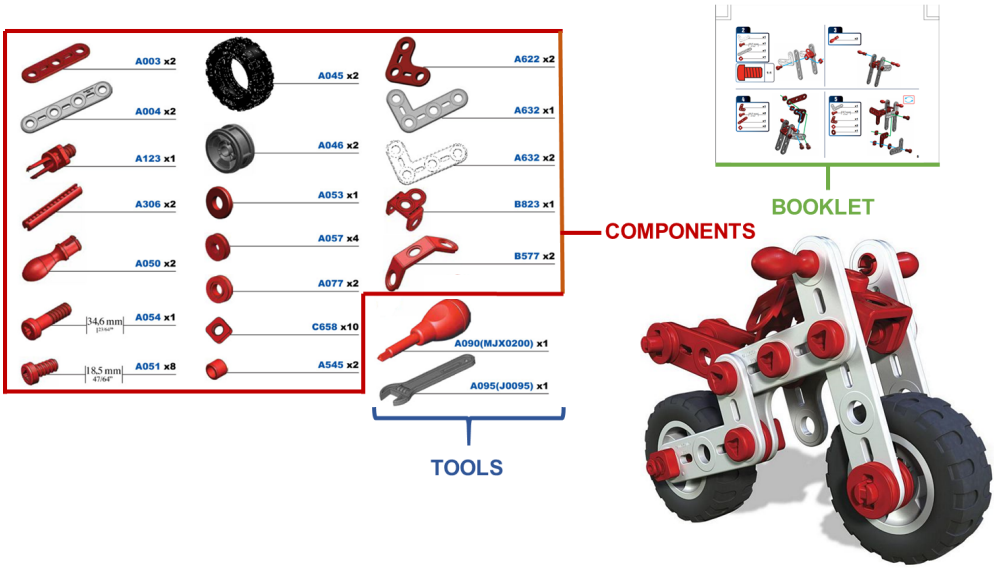

Figure 1. The toy model built by users interacting with 2 tools, 49 components and the instructions booklet. The figure is better seen on screen.

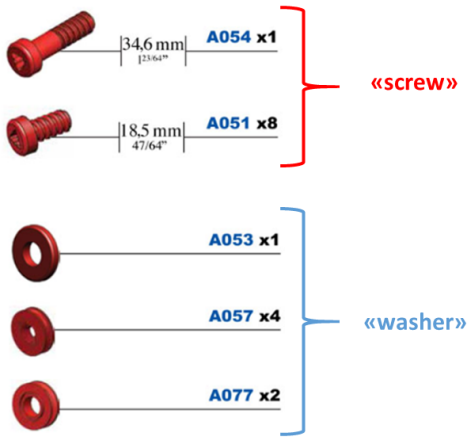

Figure 2. Grouped pieces belonging to *screw* and *washer* classes.

correct object class. Figure 5 reports an example of the 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 6 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 7. 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 8 reports all the action classes obtained from the 12 verbs classes of the MECCANO dataset as discussed in the following. Let $O = \{o_1, o_2, ..., o_n\}$ and $V = \{v_1, v_2, ..., v_m\}$ 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 $o = instruction\ booklet$. Hence, we defined the two action classes *check instruction booklet* and *browse in-*

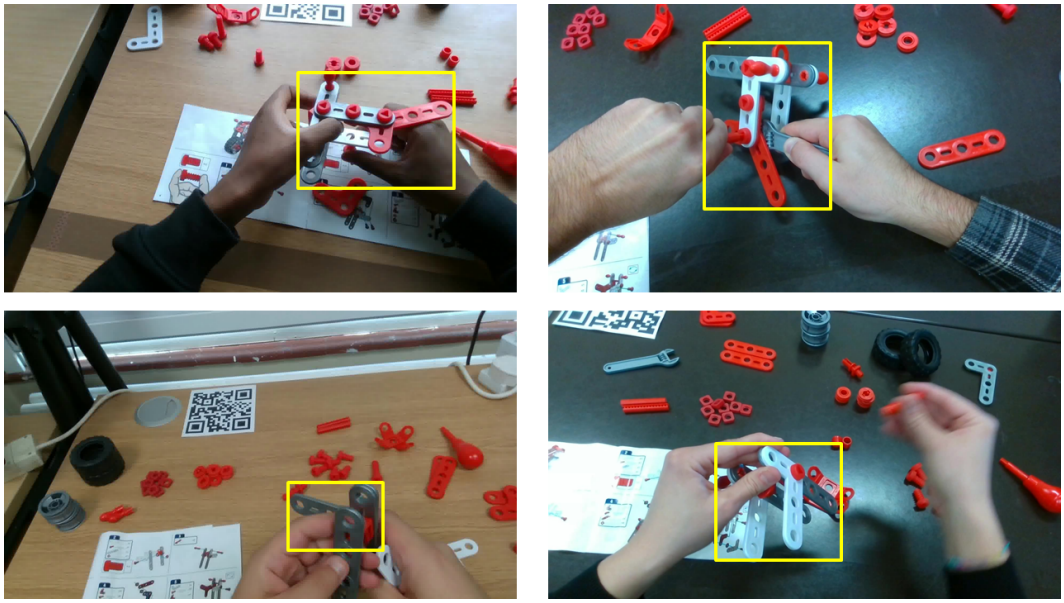

Figure 3. Examples of objects belonging to the partial model class.

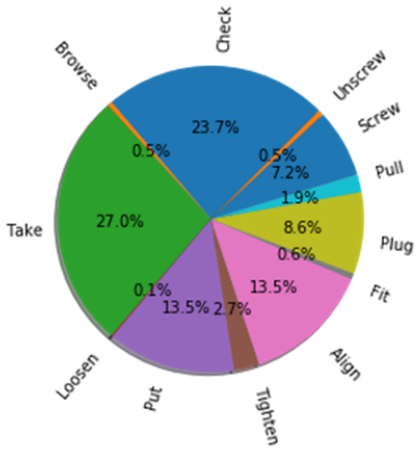

Figure 4. Fractions of instances of each verb in the MECCANO dataset.

struction 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 $v = tighten$, the object $o = bolt$ and the tool $o = wrench$. 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.

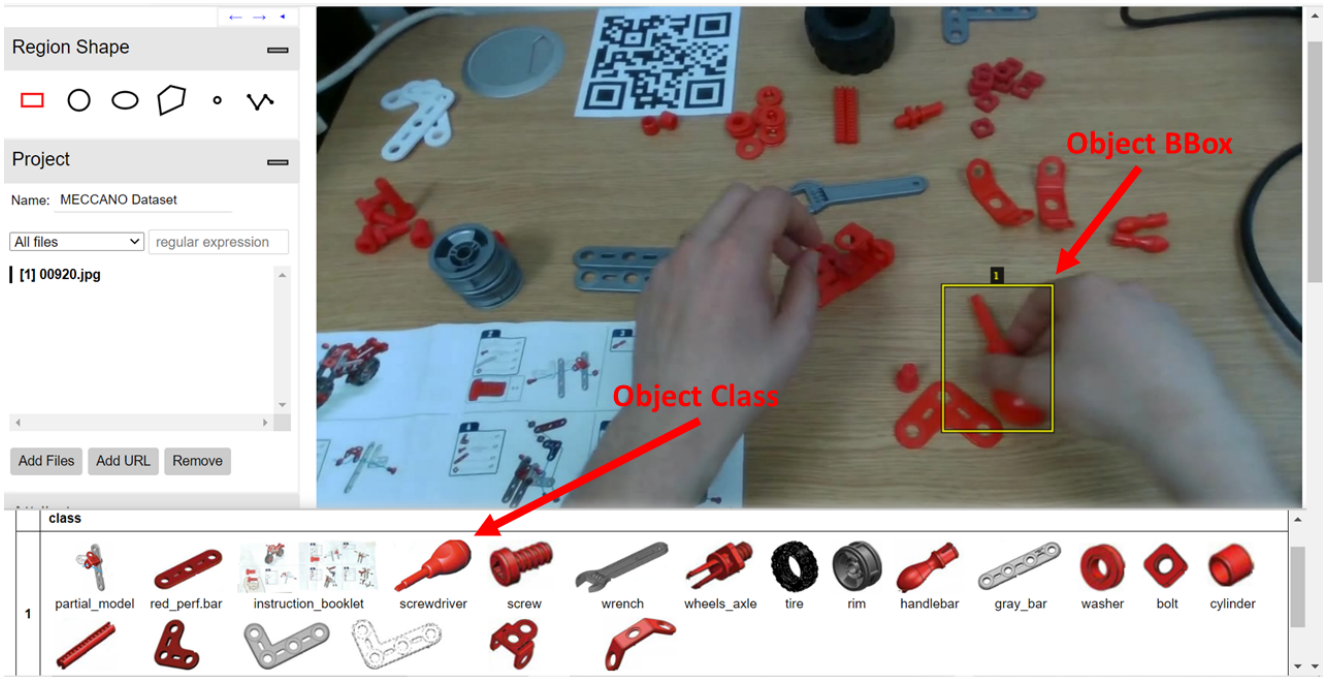

Figure 5. Customized VIA project to support the labeling of active objects. Annotators were presented with a panel which allowed them to identify object classes through their thumbnails.

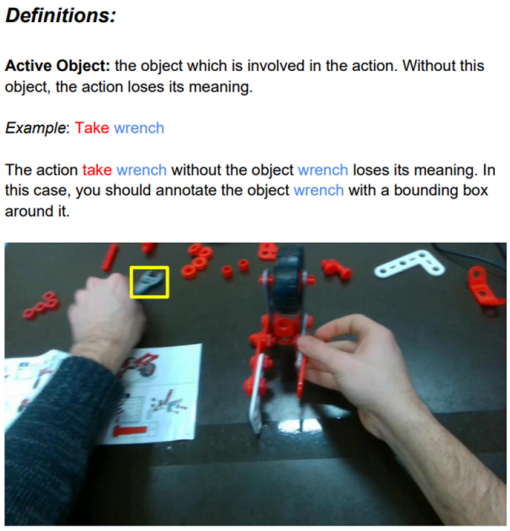

Figure 6. *Active* object definition given to the labelers for the *active* object bounding box annotation stage.

3. Baseline Implementation Details

3.1. 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.

3.2. 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 [4]. 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.

3.3. Active Object Recognition

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

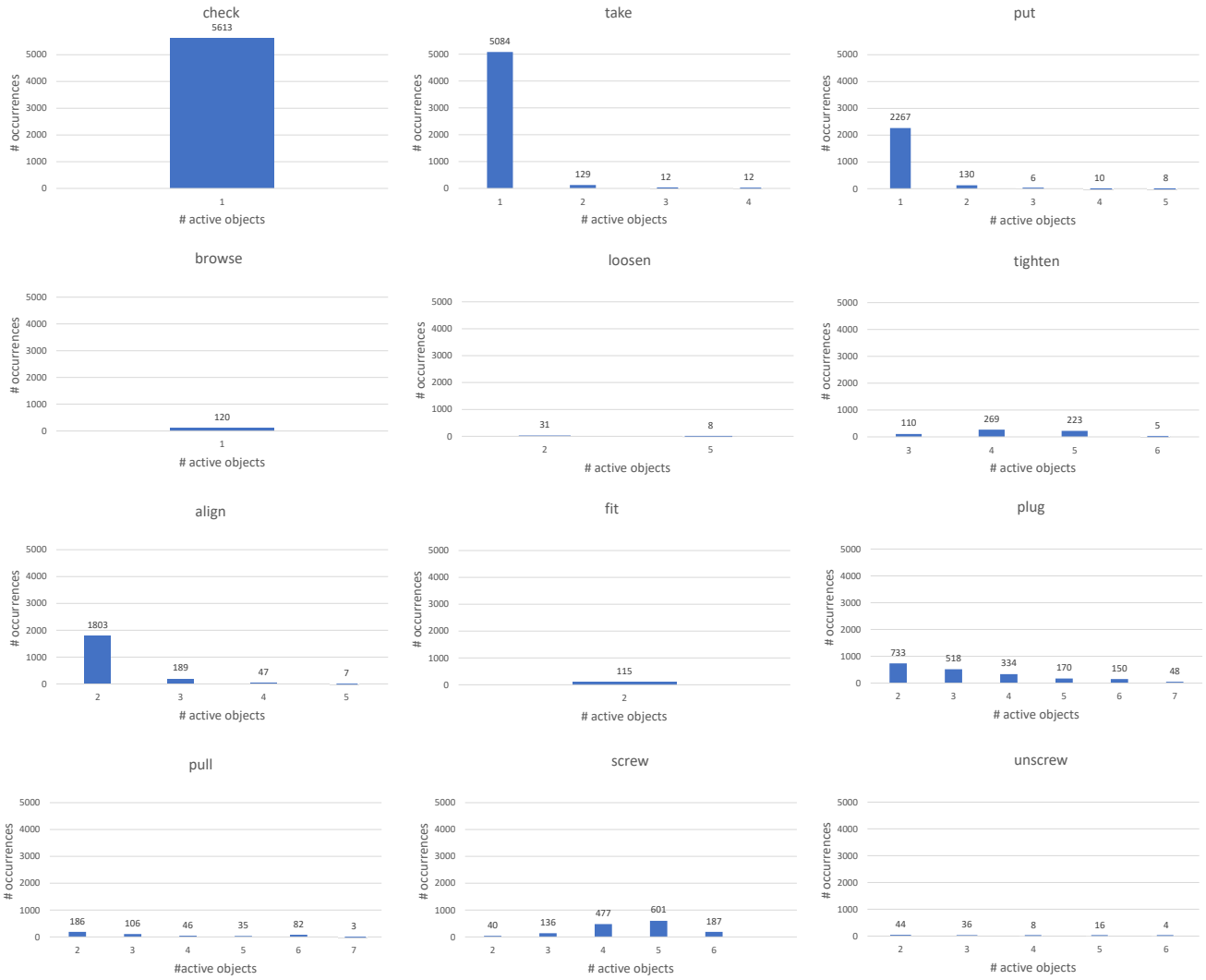

Figure 7. Number of objects and occurrences of *active* objects related to each verb.

3.4. 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 [3] 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 form 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

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%
	mAP (per video)	31.71%	33.59%	28.89%	33.47%	28.57%	28.08%	28.44%	**30.39%**

Table 1. 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.

rate equal to 0.0001. The outputs of the three MLPs are hence fused by late fusion.

4. Additional Results

Figure 9 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 1 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 10 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 10).

References

[1] 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.

[2] The Language Archive Nijmegen: Max Planck Institute for Psycholinguistics. Elan (version 5.9) [computer software]. 2020.

[3] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. 2020.

[4] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

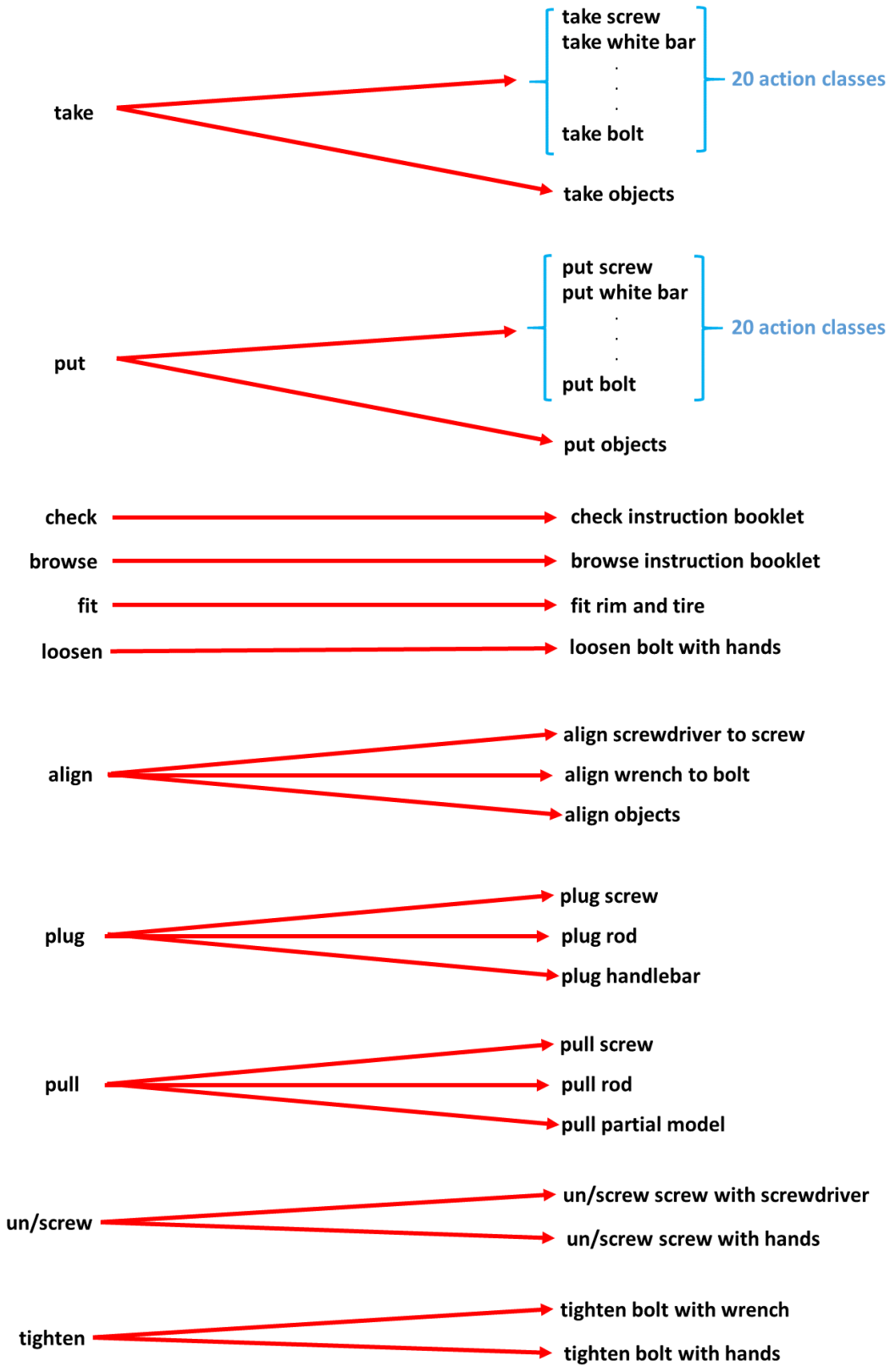

Figure 8. 61 action classes definition from the 12 verb classes and the analysis performed observing the participant behavior.

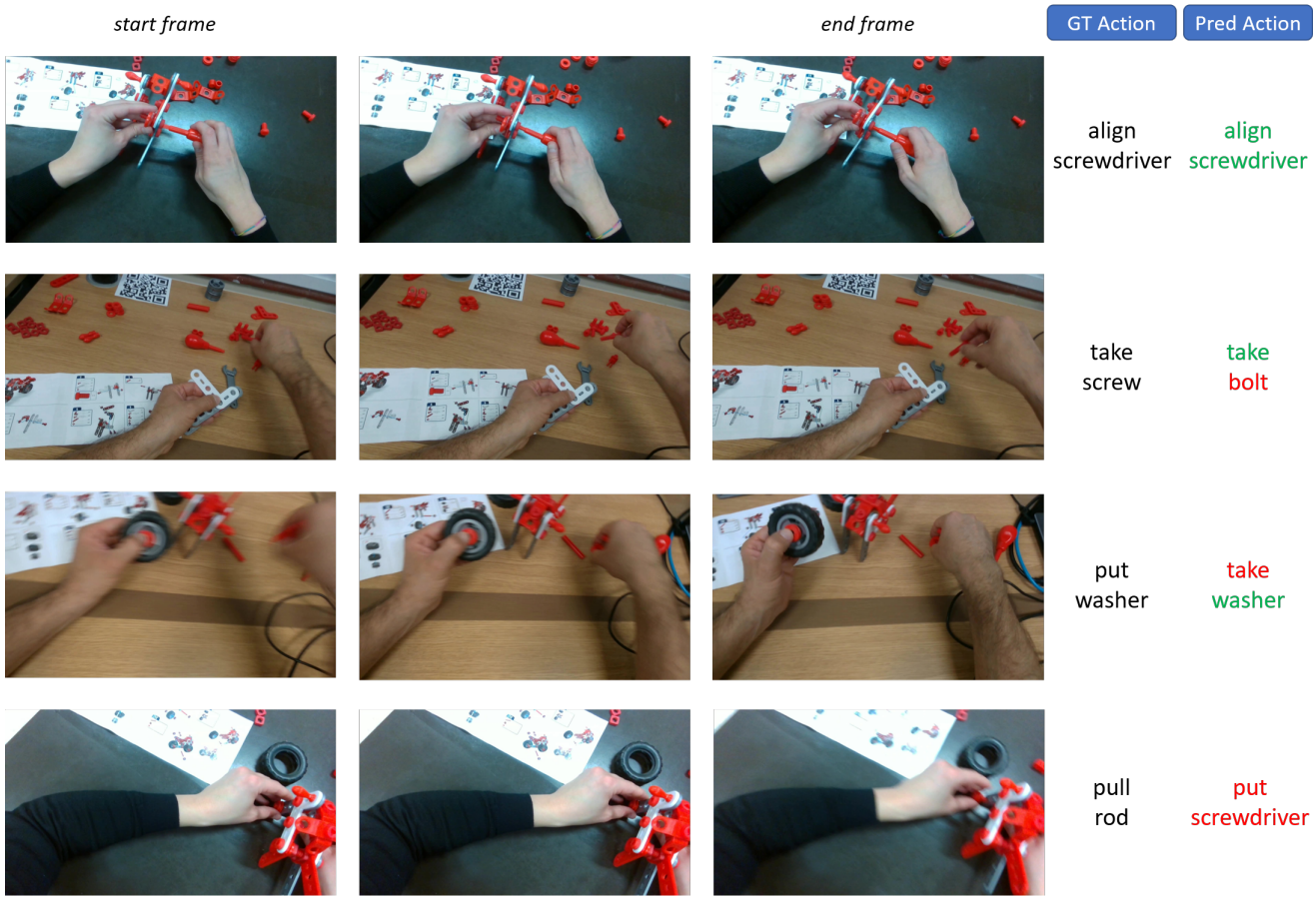

Figure 9. Qualitative results for the action recognition task. Correct predictions are in green while wrong predictions are in red.

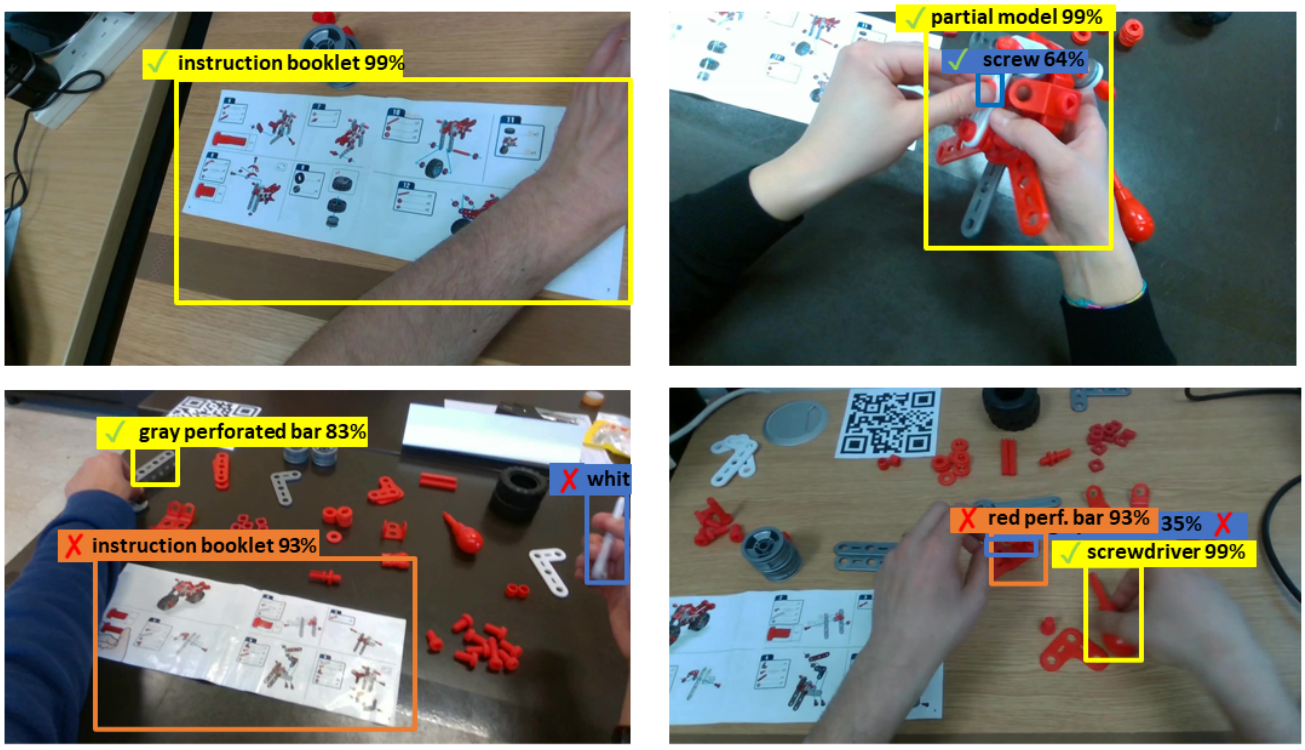

Figure 10. Qualitative results for the *active* object recognition task.