Scaling Egocentric Vision: The EPIC-KITCHENS Dataset

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray

Abstract—First-person vision is gaining interest as it offers a unique viewpoint on people's interaction with objects, their attention, and even intention. However, progress in this challenging domain has been relatively slow due to the lack of sufficiently large datasets. In this paper, we introduce EPIC-KITCHENS, a large-scale egocentric video benchmark recorded by 32 participants in their native kitchen environments. Our videos depict **nonscripted** daily activities: we simply asked each participant to start recording every time they entered their kitchen. Recording took place in 4 cities (in North America and Europe) by participants belonging to 10 different nationalities, resulting in highly diverse kitchen habits and cooking styles. Our dataset features 55 hours of video consisting of 11.5M frames, which we densely labeled for a total of 39.6K action segments and 454.2K object bounding boxes. Our annotation is unique in that we had the participants narrate their own videos (after recording), thus reflecting true intention, and we crowd-sourced ground-truths based on these. We describe our object, action and anticipation challenges, and evaluate several baselines over two test splits, *seen* and *unseen* kitchens.

Index Terms—Egocentric Vision, Dataset, Benchmarks, First-Person Vision, Egocentric Object Detection, Action Recognition and Anticipation

1 Introduction

In recent years, we have seen significant progress in many domains such as image classification [1], object detection [2], captioning [3] and visual question-answering [4]. This success has in large part been due to advances in deep learning [5] as well as the availability of large-scale image benchmarks such as Pascal VOC [6], ImageNet [7], MS-COCO [8] and ADE [9].

While gaining attention, work in video understanding has been more scarce, mainly due to the lack of annotated datasets. This has been changing recently, with the release of the action classification benchmarks such as [10], [11], [12], [13], [14], [15]. In [13], the authors collected clips from movies for the task of video-based captioning, while [14] evaluates story-based question-answering from videos. With the exception of [14], most of these datasets contain videos that are very short in duration, i.e., only a few seconds long, focusing on a single action. Hollywood in Homes [16] makes a step towards activity recognition by collecting 10K videos of humans performing various tasks in their home. While this dataset is a nice attempt to collect daily actions, the videos have been recorded in a scripted way, by asking AMT workers to act out a script in front of the camera. This makes the videos look oftentimes less natural, and they also lack the progression and multi-tasking of actions that occur in real life.

Here we focus on first-person vision, which offers a unique viewpoint on people's daily activities. This data is rich as it reflects our goals and motivation, ability to multi-task, and the many different ways to perform a variety of important, but mundane, everyday tasks (such as cleaning the dishes). Egocentric data has also recently been proven valuable for human-to-robot imitation learning [17], [18], and has a direct impact on HCI applications. However, datasets to evaluate first-person vision algorithms [19], [20], [21], [22], [23] have been significantly smaller in size than their third-person counterparts, often captured in a single environment [19], [20], [21], [22]. Daily interactions from wearable cameras are also scarcely available online, making this a largely unavailable source of information.

In this paper, we introduce EPIC-KITCHENS, a large-scale egocentric dataset. Our data was collected by 32 participants, belonging to 10 nationalities, in their native kitchens (Fig. 1). The participants were asked to capture all their daily kitchen activities, and record sequences regardless of their duration. The recordings, which include both video and sound, not only feature the typical interactions with one's own kitchenware and appliances, but importantly show the natural multi-tasking that one performs, like washing a few dishes amidst cooking. Such parallel-goal interactions have not been captured in existing datasets, making this both a more realistic as well as a more challenging set of recordings. A video introduction to the recordings is available at: http://youtu.be/Dj6Y3H0ubDw.

Our data was captured by participants belonging to 10 nationalities, which results in a diverse set of cooking styles. Altogether, EPIC-KITCHENS has 55hrs of recording, densely annotated with start/end times for each action/interaction, as well as bounding boxes around objects subject to interaction. We describe our object, action and anticipation challenges, and report baselines in two scenarios, i.e., *seen* and *unseen* kitchens. We released all our data, and plan to track the community's progress on all challenges (with held out test ground-truth) via an online leaderboard. Details at: http://epic-kitchens.github.io.

- *Authors list sorted alphabetically*
- *D. Damen, H. Doughty, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price and M. Wray are with the Department of Computer Science, University of Bristol, UK. Email: <firstname>.<lastname>@bristol.ac.uk*
- *S. Fidler is with the University of Toronto and Vector Institute, Canada. Email: fidler@cs.toronto.edu*
- *A. Furnari and G. Maria Farinella are with the University of Catania, Italy. Email: <furnari, gfarinella>@dmi.unict.it*

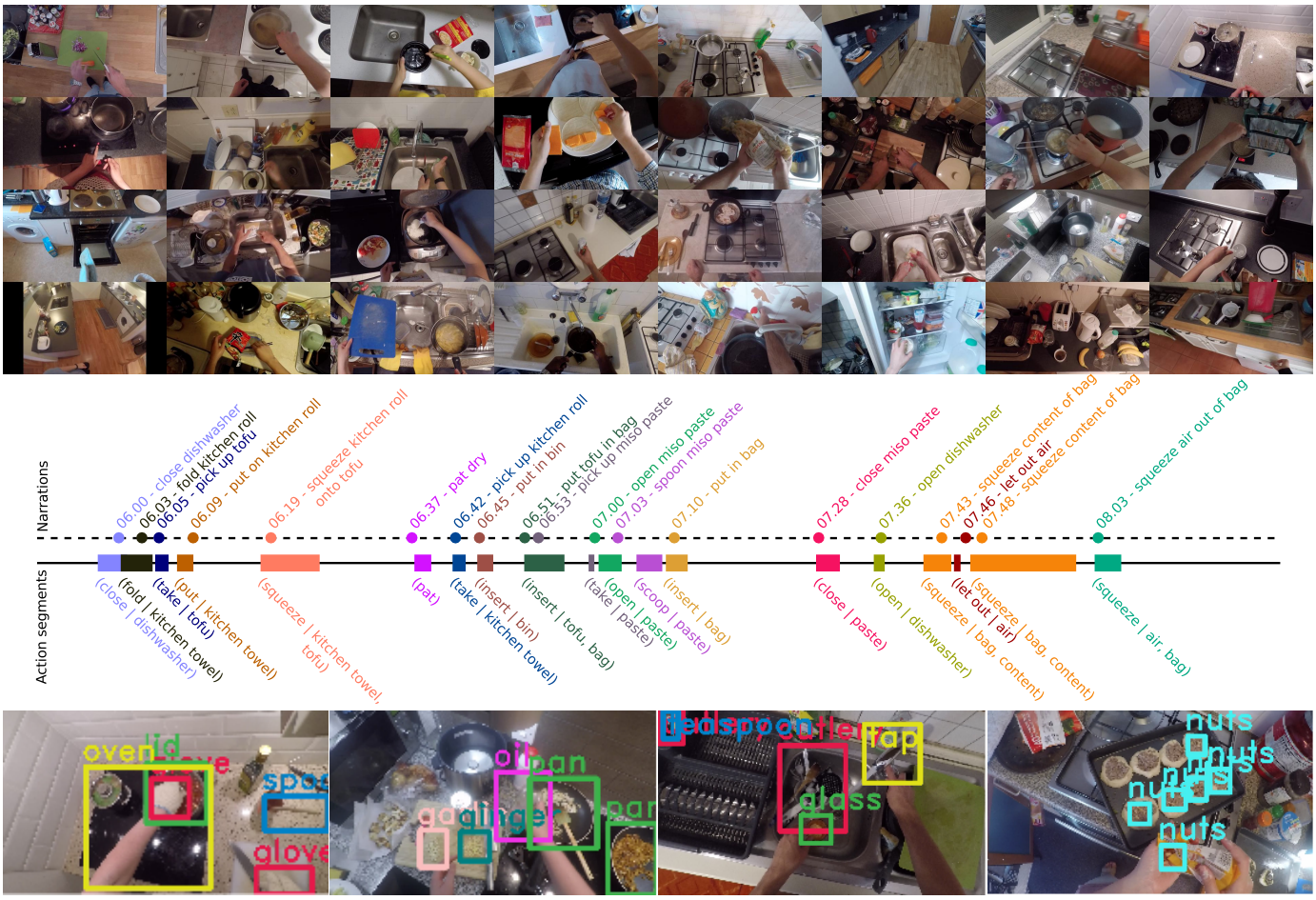

Fig. 1: **From Top**: Frames from the 32 environments; Narrations by participants used to annotate action segments; Active object bounding box annotations

TABLE 1: Comparative overview of relevant datasets. *action classes with > 50 samples

Dataset	Ego?	Non-Scripted?	Native Env?	Year	Frames	Sequences	Action Segments	Action Classes	Object BBs	Object Classes	Participants	No. Env.s
EPIC-KITCHENS	✓	✓	✓	2018	11.5M	432	39,596	149*	454,158	323	32	32
EGTEA Gaze+ [19]	✓	×	×	2018	2.4M	86	10,325	106	0	0	32	1
BEOID [21]	✓	×	×	2014	0.1M	58	1,488	34	0	0	5	1
GTEA Gaze+ [20]	✓	×	×	2012	0.4M	35	3,371	42	0	0	13	1
ADL [23]	✓	×	✓	2012	1.0M	20	436	32	137,780	42	20	20
CMU [22]	✓	×	×	2009	0.2M	16	516	31	0	0	16	1
VLOG [15]	×	✓	✓	2017	37.2M	114K	0	0	0	0	10.7K	N/A
Charades [16]	×	×	✓	2016	7.4M	9,848	67,000	157	0	0	N/A	267
Breakfast [24]	×	✓	✓	2014	3.0M	433	3078	50	0	0	52	18
50 Salads [25]	×	×	×	2013	0.6M	50	2967	52	0	0	25	1
MPII Cooking 2 [26]	×	×	×	2012	2.9M	273	14,105	88	0	0	30	1

2 Related Datasets

We compare EPIC-KITCHENS to five commonly-used egocentric datasets [19], [20], [21], [22], [23] in Table 1, as well as five third-person activity-recognition datasets [15], [16], [24], [25], [26] that focus on object-interaction activities. We exclude egocentric datasets that focus on inter-person interactions [27], [28], [29] as well as instructional videos [30], [31], as these target a different research question.

A few datasets aim at capturing activities in native environments, most of which are recorded in third-person [10], [15], [16], [24]. [24] focuses on cooking dishes based on a list of breakfast recipes. In [15], short segments linked to interactions with 30 daily objects are collected by querying YouTube, while [10], [16] are scripted – subjects are requested to enact a crowd-sourced storyline [16] or a given action [10], which oftentimes results in less natural looking actions. Most egocentric datasets similarly use scripted activities, i.e. people are told what actions to perform. When following instructions, participants perform steps in a sequential order, as opposed to the more natural real-life scenarios addressed in our work, which involve multi-tasking, searching for an item, thinking what to do next, changing one's mind or even unexpected surprises. EPIC-KITCHENS is most closely related to the ADL dataset [23] which also provides egocentric recordings in

native environments. However, our dataset is substantially larger: it has 11.5M frames vs 1M in ADL, 90x more annotated action segments, and 4x more object bounding boxes, making it the largest first-person dataset to date.

3 THE EPIC-KITCHENS DATASET

In this section, we describe our data collection and annotation pipeline. We also present various statistics, showcasing different aspects of our collected data.

3.1 Data Collection

The dataset was recorded by 32 individuals in 4 cities in different countries (in North America and Europe): 15 in Bristol/UK, 8 in Toronto/Canada, 8 in Catania/Italy and 1 in Seattle/USA between May and Nov 2017. Participants were asked to capture all kitchen visits *for three consecutive days*, with the recording starting immediately before entering the kitchen, allowing a few seconds to ensure the camera starts before carrying out the daily activities. and only stopped before leaving the kitchen. They recorded the dataset voluntarily and were not financially rewarded. The participants were asked to be in the kitchen alone for all the recordings, thus capturing only one-person activities. We also asked them to remove all items that would disclose their identity such as portraits or mirrors.

Data was captured using a head-mounted Go-Pro with an adjustable mounting to control the viewpoint for different environments and participants' heights. Before each recording, the participants checked the battery life and viewpoint, using the GoPro Capture mobile app, so that their stretched hands were approximately located at the middle of the camera frame. The camera was set to linear field of view, 59.94*fps* and Full HD resolution of 1920x1080, however some subjects made minor changes like wide or ultra-wide fov or resolution, as they recorded multiple sequences in their homes, and thus were switching the device off and on over several days. Specifically, 1% of the videos were recorded at 1280x720 and 0.5% at 1920x1440. Also, 1% at 30*fps* , 1% at 48*fps* and 0.2% at 90*fps* .

The recording lengths varied depending on the participant's kitchen engagement. On average, people recorded for 1.7hrs, with the maximum being 4.6hrs and the minimum just over half an hour. Cooking a single meal can span multiple sequences, depending on whether one stays in the kitchen, or leaves and returns later.

Fig. 2: Head-mounted GoPro used in dataset recording

Use any word you prefer. Feel free to vary your words or stick to a few.
Use present tense verbs (e.g. cut/open/close).
Use verb-object pairs (e.g. wash carrot).
You may (if you prefer) skip articles and pronouns (e.g. "cut kiwi" rather than "I cut the kiwi").
Use propositions when needed (e.g. "pour water into kettle").
Use 'and' when actions are co-occurring (e.g. "hold mug and pour water").
If an action is taking long, you can narrate again (e.g. "still stirring soup").

Fig. 3: Instructions used to collect video narrations from our participants

On average, each participant recorded 13.6 sequences. Figure 4 presents statistics on time of day using the local-time of the recording, high-level goals and sequence durations.

Since crowd-sourcing annotations for such long videos is very challenging, we had our original participants do a coarse first annotation. Each participant was asked to watch their videos, after completing all recordings, and narrate the actions carried out, using a hand-held recording device. We opted for a sound recording rather than written captions as this is arguably much faster for the participants, who were thus more willing to provide these annotations[1]. These are analogous to a *live commentary* of the video. The general instructions for narrations are listed in Fig. 3. The participant narrated in English if sufficiently fluent or in their native language. In total, 5 languages were used: 17 narrated in English, 7 in Italian, 6 in Spanish, 1 in Greek and 1 in Chinese. Figure 4 shows wordles of the most frequent words in each language.

Our decision to collect narrations from the participants themselves is because they are the most qualified to label the activity compared to an independent observer, as they were the ones performing the actions. We opted for a post-recording narration such that the participant performs her/his daily activities undisturbed, without being concerned about labeling.

We tested several automatic audio-to-text APIs [32], [33], [34], which failed to produce accurate transcriptions as these expect a relevant corpus and complete sentences for context. We thus collected manual transcriptions via Amazon Mechanical Turk (AMT), and used the YouTube's automatic closed caption alignment tool to produce accurate timings. For non-English narrations, we also asked AMT workers to translate the sentences. To make the job more suitable for AMT, narration audio files are split by removing silence below a pre-specified decibel threshold (after compression and normalisation). Speech chunks are then combined into HITs with a duration of around 30 seconds each. To ensure consistency, we submit the same HIT three times and select the ones with an edit distance of 0 to at least one other HIT. We manually corrected cases when there was no agreement. Examples of transcribed and timed narrations are provided in Table 2. The participants were also asked to provide one sentence per sequence describing the overall goal or activity that took place (examples in Table 3).

In total, reporting translated or originally-English narrations, we collected $39,596$ action narrations, corresponding to a narration every $4.9s$ in the video. The average number of words per phrase is 2.8 words. These narrations give us an initial labeling of all actions with rough temporal alignment (obtained from

1. A freely-available application on smart phone was used to gather the narrations' recordings.

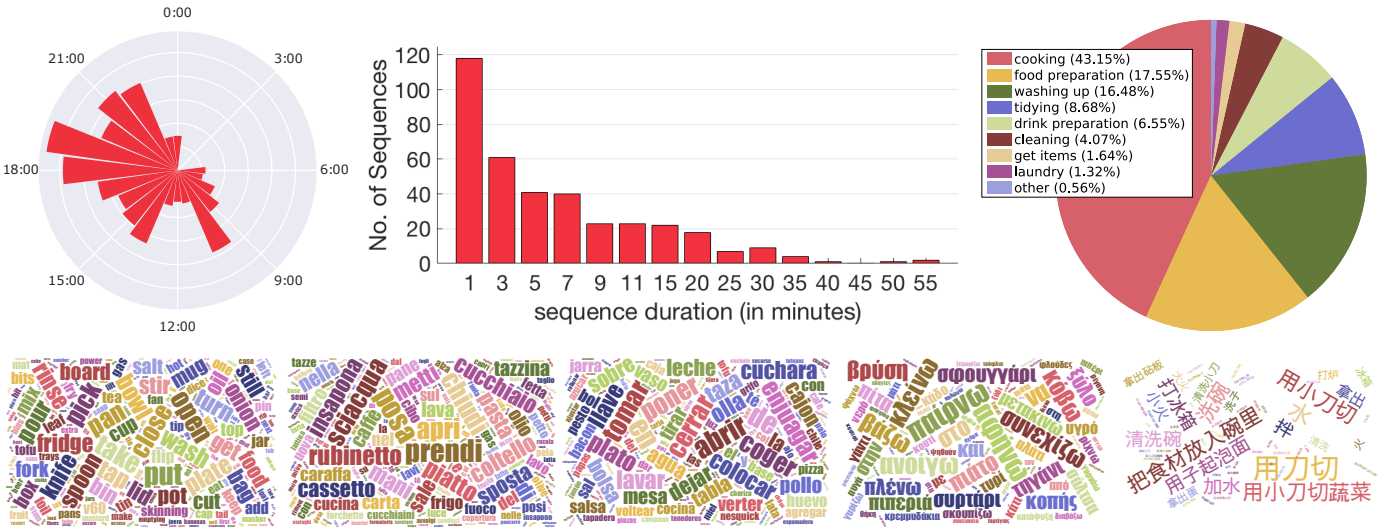

Fig. 4: **Top** (left to right): time of day of the recording, histogram of sequence durations and pie chart of high-level goals; **Bottom**: Wordles of narrations in native languages (English, Italian, Spanish, Greek and Chinese)

TABLE 2: Extracts from 6 transcription files in .sbv format

0:14:44.190,0:14:45.310 pour tofu onto pan 0:14:45.310,0:14:49.540 put down tofu container 0:14:49.540,0:15:02.690 stir vegetables and tofu 0:15:02.690,0:15:06.260 put down spatula 0:15:06.260,0:15:07.820 take tofu container 0:15:07.820,0:15:10.040 throw something into the bin	0:00:02.780,0:00:04.640 open the bin 0:00:04.640,0:00:06.100 pickup the bag 0:00:06.100,0:00:09.530 tie the bag 0:00:09.530,0:00:10.610 tie the bag again 0:00:10.610,0:00:14.309 pickup bag 0:00:14.309,0:00:17.520 put bag down	0:04:37.880,0:04:39.620 Take onion 0:04:39.620,0:04:48.160 Cut onion 0:04:48.160,0:04:49.160 Peel onion 0:04:49.160,0:04:51.290 Put peel in bin 0:04:51.290,0:05:06.350 Peel onion 0:05:06.350,0:05:15.200 Put peel in bin	0:06:40.669,0:06:41.669 pick up spatula 0:06:41.669,0:06:45.250 stir potatoes 0:06:45.250,0:06:46.250 put down spatula 0:06:46.250,0:06:50.830 turn down hob 0:06:50.830,0:06:55.819 pick up pan 0:06:55.819,0:06:57.170 tip out paneer	0:12:28.000,0:12:28.000 pour pasta into container 0:12:33.000,0:12:33.000 take jar of pesto 0 :12:39.000,0:12:39.000 take teaspoon 0:12:41.000,0:12:41.000 pour pesto in container 0:12:55.000,0:12:55.000 place pesto bottle on table 0:12:58.000,0:12:58.000 take wooden spoon	0:00:03.280,0:00:06.000 open fridge 0:00:06.000,0:00:09.349 take milk 0:00:09.349,0:00:10.910 put milk 0:00:10.910,0:00:12.690 open cupboard 0:00:12.690,0:00:15.089 take bowl 0:00:15.089,0:00:18.080 open drawer

TABLE 3: Sample Video Summaries

P04-04.avi	making curries - fried paneer, boiled potatoes, chopped veg	P07-08.avi	pour coffee and prepare tortilla with cheese and pepperoni
P13-08.avi	clean the dishes and prepare spaghetti carbonara	P19-04.avi	made steamed noodles with beans, tomatoes, chicken
P23-02.avi	cooking Indian egg curry while cleaning dishes	P28-09.avi	prepare avocado and tomato salad

the timestamp of the audio narration with respect to the video). However, narrations are also not a perfect source of ground-truth:

- The narrations can be incomplete, i.e., the participants were selective in which actions they chose to narrate. We noticed that they labeled the 'open' actions more than their counter-action 'close', as the narrator's attention has already moved to the next goal. We consider this phenomena in our evaluation, by only evaluating actions that have been narrated.
- Temporally, the narrations are belated, after the action takes place. This is adjusted using ground-truth action segments (see Sec. 3.2).
- Participants use their own vocabulary and free language. While this is a challenging issue to deal with in evaluation, we believe it is important to push the community to go beyond the pre-selected list of labels in the future (also argued in [9]). We here resolve this issue by grouping verbs and nouns into minimally overlapping classes (see Sec. 3.4).

3.2 Action Segment Annotations

For each narrated sentence, we adjust the start and end times of the action using AMT. To ensure the annotators are trained to perform temporal localization, we use a clip from our previous work's understanding [35] that explains temporal bounds of actions. Each HIT is composed of a maximum of 10 consecutive narrated phrases p_i, where annotators label $A_i = [t_{s_i}, t_{e_i}]$ as the start and end times of the i^{th} action. Two constraints were added to decrease the amount of noisy annotations: (1) action has to be at least 0.5 seconds in length; (2) action cannot start before the preceding action's start time. Note that consecutive actions are allowed to overlap. Moreover, the annotators could indicate that the action does not appear in the video. This handles occluded, impossible to distinguish or out-of-bounds cases.

To ensure consistency, we ask $\mathcal{K}_a = 4$ annotators to annotate each HIT. We filter, reject and resubmit unacceptable hits through a combination of automatic and manual checks. We then calculate the agreement for each annotation $A_i(j)$ (i is the action and j indexes the annotator) as follows:

$$\alpha_i(j) = \frac{1}{K_a} \sum_{k=1}^{\mathcal{K}_a} \text{IoU}(A_i(j), A_i(k)) \quad (1)$$

We first find the annotator with the maximum agreement $\hat{j} = \arg\max_j \alpha_i(j)$. We also find $\hat{k} = \arg\max_k \text{IoU}(A_i(\hat{j}), A_i(k))$. The ground-truth action

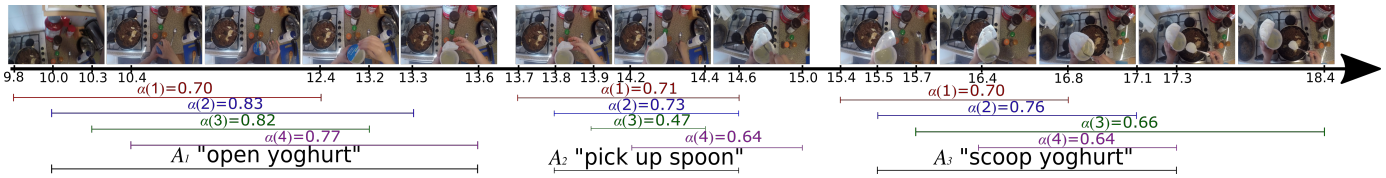

Fig. 5: An example of annotated action segments for 2 consecutive actions

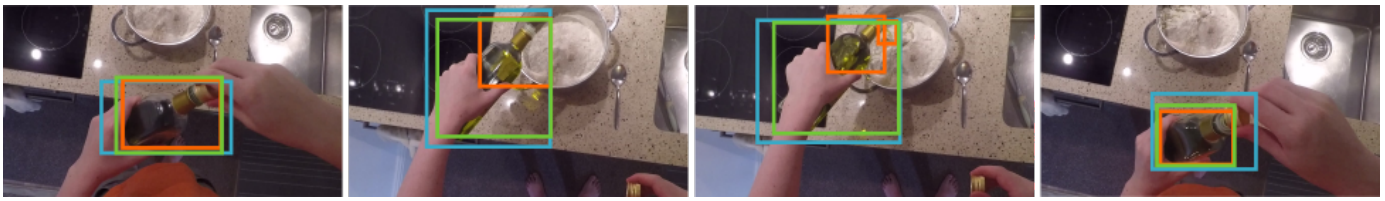

Fig. 6: Object annotation from three AMT workers (orange, blue and green). The green participant's annotations are selected as the final annotations

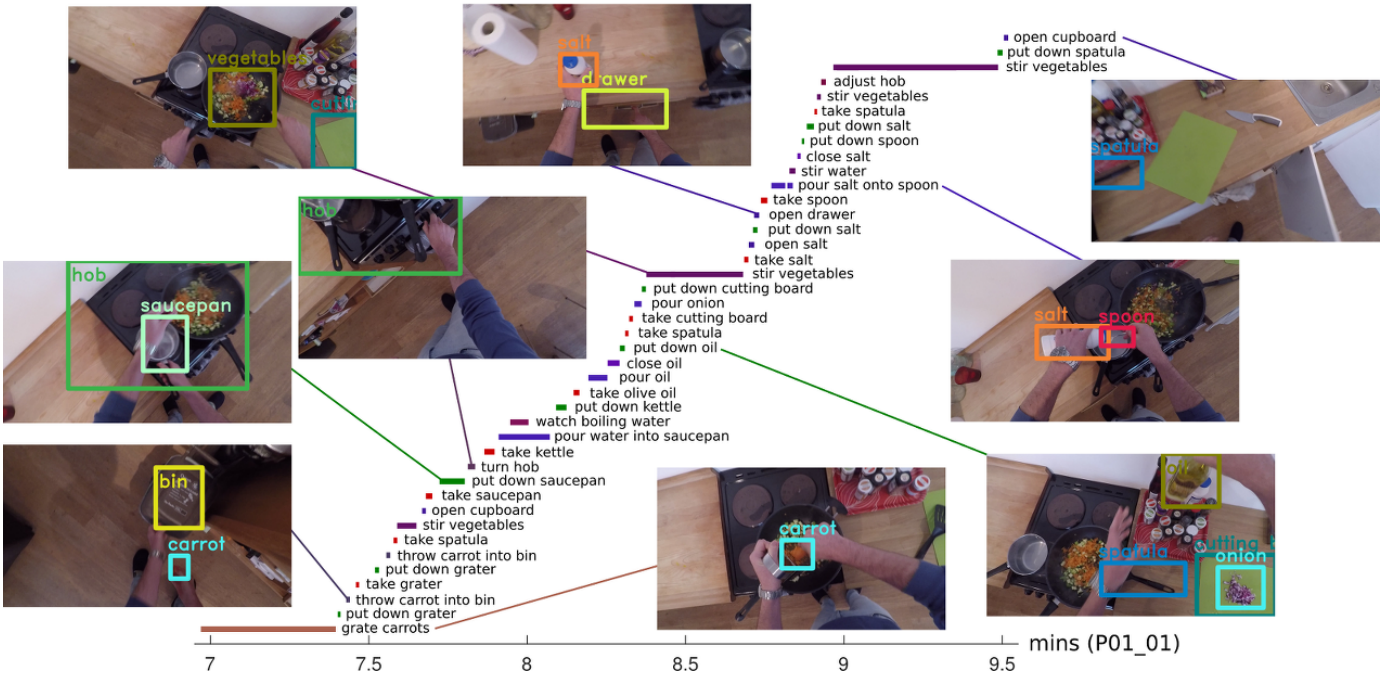

Fig. 7: Sample consecutive action segments with keyframe object annotations

segment A_i is then defined as:

$$A_i = \begin{cases} \text{Union}(A_i(\hat{j}), A_i(\hat{k})), & \text{if IoU}(A_i(\hat{j}), A_i(\hat{k})) > 0.5 \\ A_i(\hat{j}), & \text{otherwise} \end{cases} \quad (2)$$

We thus combine two annotations when they have a strong agreement, since in some cases the single (best) annotation results in a too tight of a segment. Figure 5 shows examples of combining annotations.

In total, we collected such labels, which we call the strong action labels, for 39,564 action segments (lengths: $\mu = 3.7s$, $\sigma = 5.6s$). These represent 99.9% of all narrations. The missed annotations were those labeled as "not visible" by the annotators, though mentioned in narrations.

3.3 Active Object Bounding Box Annotations

The narrated *nouns* correspond to objects relevant to the action [21], [36]. Assume $\mathcal{O}_i$ is the set of one or more nouns in the phrase p_i associated with the action segment $A_i = [t_{s_i}, t_{e_i}]$. We consider each frame f within $[t_{s_i} - 2s, t_{e_i} + 2s]$ as a potential frame to annotate the bounding box(es), for each object in $\mathcal{O}_i$. We build on the interface from [37] for annotating bounding boxes on AMT. Each HIT aims to get an annotation for one object, for the maximum duration of $25s$, which corresponds to 50 consecutive frames at $2fps$. The annotator can also note that the object does not exist in f. We particularly ask the same annotator to annotate consecutive frames to avoid subjective decisions on the extents of objects. We also assess annotators' quality by ensuring that the annotators obtain an IoU ≥ 0.7 on two golden annotations at the start of every HIT. We request $\mathcal{K}_o = 3$ workers per HIT, and select the one with maximum agreement β:

$$\beta(q) = \sum_{f} \max_{j \neq q}^{\mathcal{K}_o} \max_{k,l} \text{IoU}(\text{BB}(j, f, k), \text{BB}(q, f, l)) \quad (3)$$

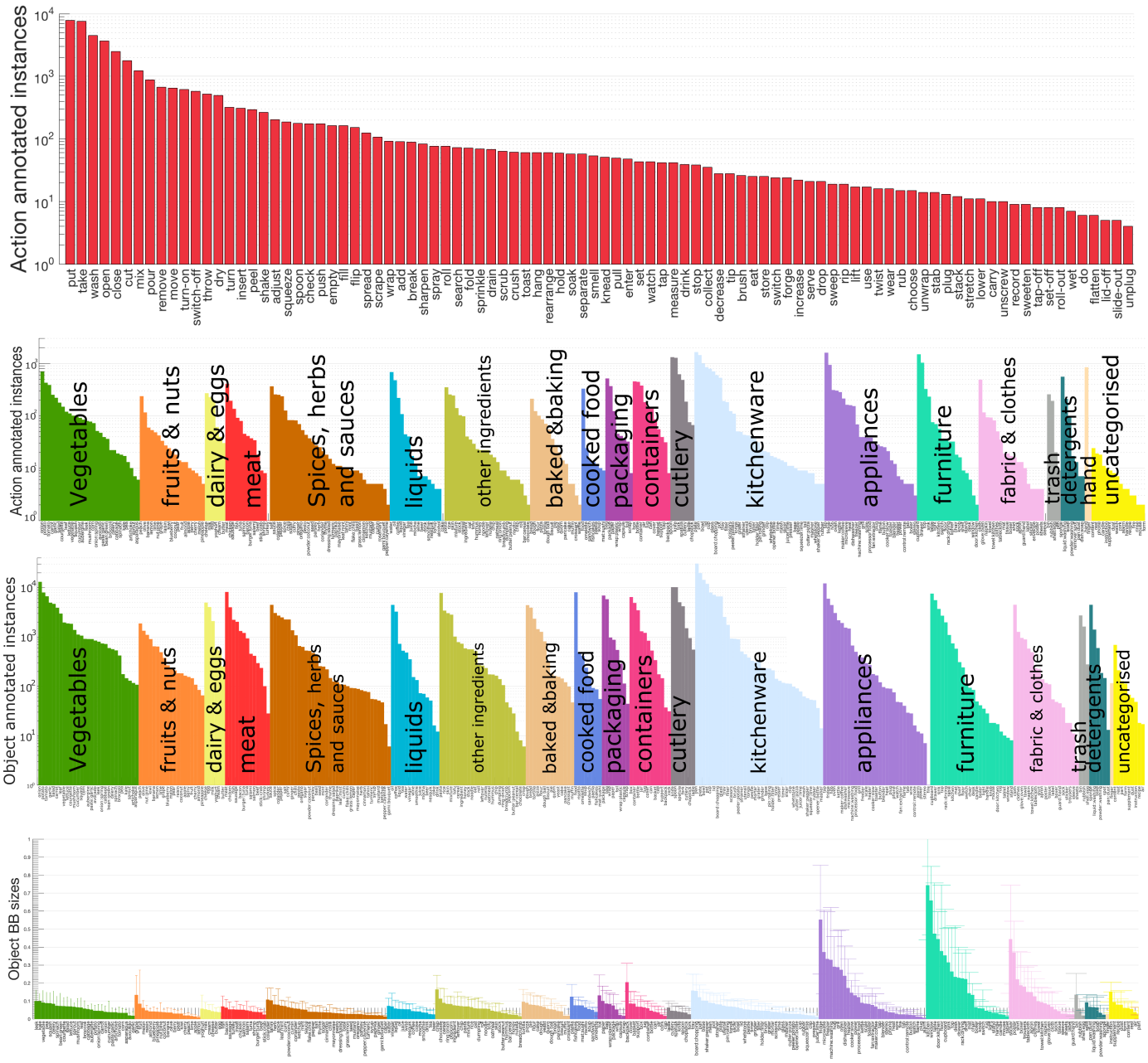

Fig. 8: **From Top**: Frequency of verb classes in action segments; Frequency of noun clusters in action segments, by category; Frequency of noun clusters in bounding box annotations, by category; Mean and standard deviation of bounding box, by category

where $\text{BB}(q, f, k)$ is the k^{th} bounding box annotation by annotator q in frame f. Ties are broken by selecting the worker who provides the tighter bounding boxes. Figure 6 shows multiple annotations for four keyframes in a sequence.

In total, we collected 454,158 bounding boxes (per frame: $\mu = 1.64$ boxes, $\sigma = 0.92$). Sample action segments and object bounding boxes are shown in Fig. 7.

3.4 Verb and Noun Classes

Since our participants annotated using free text in multiple languages, a variety of verbs and nouns have been collected. For example, 'put', 'place', 'put-down', 'put-back', 'leave' or 'return' have all been used to indicate putting an object in a certain location. We attempt to group these into classes with minimal semantic overlap, to accommodate the more typical approaches to multi-class detection and recognition where each example is believed to belong to one class only. We estimate Part-of-Speech (POS), using spaCy's English core web model, to determine the verbs and nouns in the phrase. This was necessary as although the majority of annotations are verb-noun phrases, such as 'take cup' or 'open fridge', there were annotations which included prepositions such as 'put pan on hob' as well as annotations which included multiple objects such as 'put down onion and knife'. We find the verb by selecting the first verb in the sentence, and find all nouns in the sentence excluding any that match the chosen verb. When a noun is absent or replaced by a pronoun (*e.g. 'it'*), we use the noun from the directly preceding narration (e.g. p_i: 'rinse cup', p_{i+1}: 'place it to dry').

TABLE 4: Statistics of test splits: seen (S1) and unseen (S2) kitchens

	#Subjects	#Sequences	Duration (s)	%	Narrated Segments	Action Segments	Bounding Boxes
Train/Val	28	272	141731		28,588	28,561	326,298
S1 Test	28	106	39084	20%	8,069	8,064	97,865
S2 Test	4	54	13231	7%	2,939	2,939	29,995

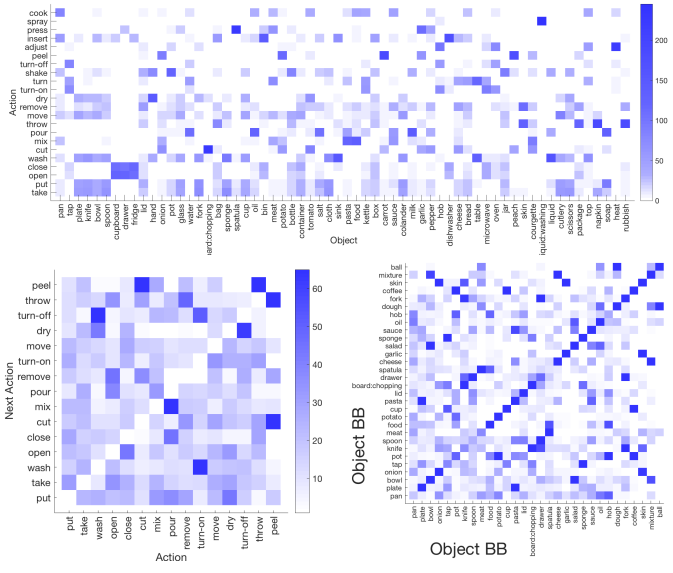

Fig. 9: **Top**: Frequently co-occurring verb/nouns in action segments [e.g. (open/close, cupboard/drawer/fridge), (peel, carrot/onion/potato/peach), (adjust, heat)]; **Bottom Left**: Next-action excluding repetitive instances of the same action [e.g. peel → cut, turn-on → wash, pour → mix].; **Bottom Right**: Co-occurring bounding boxes in one frame [e.g. (pot, coffee), (knife, chopping board), (tap, sponge)]

We refer to the set of minimally-overlapping verb classes as C_V, and similarly C_N for nouns. We attempted to automate the clustering of verbs and nouns using combinations of WordNet [38], Word2Vec [39], and Lesk algorithm [40], however, due to limited context there were too many meaningless clusters. We thus elected to manually cluster the verbs and semi-automatically cluster the nouns. We preprocessed the compound nouns *e.g.* 'pizza cutter' as a subset of the second noun *e.g.* 'cutter'. We then manually adjusted the clustering, merging the variety of names used for the same object, *e.g.* 'cup' and 'mug', as well as splitting some base nouns, *e.g.* 'washing machine' vs 'coffee machine'.

In total, we have 125 C_V classes and 331 C_N classes. In Fig. 8, we show C_V ordered by frequency of occurrence in action segments, C_N ordered by frequency of occurrence in action segments as well as C_N ordered by number of annotated bounding boxes. These are grouped into 19 super categories, of which 9 are food and drinks, with the rest containing kitchen essentials from appliances to cutlery. The figure also shows the sizes of the annotated bounding boxes for these categories. Co-occurring classes are presented in Fig. 9.

3.5 Annotation Quality Assurance

To analyse the quality of annotations, we choose 300 random samples, and manually assess correctness. We report:

- **Action Segment Boundaries** (A_i): We check that the start/end times fully enclose the action boundaries, with any additional frames not part of other actions - error: *5.7%*.
- **Object Bounding Boxes** ($\mathcal{O}_i$): We check that the bounding box encapsulates the object or its parts, with minimal overlap with other objects, and that all instances of the class in the frame have been labeled – error: *6.3%*.
- **Verb classes** (C_V)**:** We check that the verb class is correct – error: *3.3%*.
- **Noun classes** (C_N)**:** We check that the noun class is correct – error : *6.0%*.

These error rates are comparable to recently published datasets [12].

4 Benchmarks and Baseline Results

EPIC-KITCHENS offers a variety of potential challenges from routine understanding, to activity recognition and object detection. As a start, we define three challenges for which we provide baseline results. For the evaluation protocols, we hold out ground truth annotations for 27% of the data (Table 4). We particularly aim to assess the generalizability to novel environments, and we thus structured our test set to have a collection of *seen* and previously *unseen* kitchens:

Seen Kitchens (S1): In this split, each kitchen is seen in both training and testing, where roughly 80% of sequences are in training and 20% in testing. We do not split sequences, thus each sequence is in either training or testing.

Unseen Kitchens (S2): This divides the participants/kitchens so all sequences of the same kitchen are either in training or testing. We hold out the complete sequences for 4 participants for this testing protocol. The test set of S2 is only 7% of the dataset in terms of frame count, but the challenges remain considerable.

Appendices A and B contain the sequences in the two test splits.

We now evaluate several existing methods on our benchmarks, to gain an understanding of how challenging our dataset is.

4.1 Object Detection Benchmark

Challenge: This challenge focuses on object detection for all of our C_N classes. Note that our annotations only capture the 'active' objects pre-, during- and post- interaction. We thus restrict the images evaluated per class to those where the object has been annotated. We particularly aim to break the performance down into multi-shot and few-shot class groups, so as to analyze the capabilities of the approaches to quickly learn novel objects (with only a few examples). Our challenge leaderboard will reflect the methods' abilities on both sets of classes.

Method: We evaluate object detection using Faster R-CNN [2] due to its state-of-the-art performance. Faster R-CNN uses a region proposal network (RPN) to first generate class agnostic object proposals, and then classifies these and outputs refined bounding box predictions. We use the implementation from [41],

TABLE 5: Baseline results for the Object Detection challenge

		15 Most Frequent Object Classes															Totals		
	mAP	pan	plate	bowl	onion	tap	pot	knife	spoon	meat	food	potato	cup	pasta	cupboard	lid	few-shot	many-shot	all
S1	IoU > 0.05	74.00	72.61	71.50	60.72	84.44	69.97	44.03	40.93	29.65	58.52	62.82	53.30	78.39	51.95	62.77	9.71	49.80	38.23
	IoU > 0.5	67.60	66.21	65.98	39.96	73.80	64.71	28.80	23.89	20.75	49.85	55.48	42.99	69.75	29.20	58.48	6.98	36.50	28.06
	IoU > 0.75	21.94	44.60	39.48	3.52	25.83	19.67	3.42	2.59	5.27	15.78	13.18	8.00	24.53	4.05	26.51	0.36	8.73	6.50
S2	IoU > 0.05	75.94	87.36	72.72	47.61	78.14	75.92	55.51	41.28	71.59	38.61	N/A	44.62	80.58	53.88	58.40	6.00	51.71	40.61
	IoU > 0.5	62.88	84.86	68.61	32.18	59.75	62.86	39.60	27.52	53.54	35.47	N/A	39.19	76.27	32.54	49.36	5.32	36.27	28.57
	IoU > 0.75	14.56	62.82	38.44	2.25	4.89	14.91	3.85	1.51	9.56	8.10	N/A	7.60	43.30	5.61	25.48	0.18	9.05	7.04

Fig. 10: Qualitative results for the object detection challenge

[42] with a base architecture of ResNet-101 [1] pretrained on MS-COCO [8].

Implementation Details: Learning rate is initalised to 0.0003 decaying by a factor of 10 after 30000 and 40000 iterations. We use a minibatch size of 4 on 8 Nvidia P100 GPUs on a single compute node (Nvidia DGX-1) with distributed training and parameter synchronisation – i.e. overall minibatch size of 32. As in [2], images are rescaled such that their shortest side is 600 pixels and the aspect ratio is maintained. We use a stride of 16 on the last convolution layer for feature extraction and for anchors we use 4 scales of 0.25, 0.5, 1.0 and 2.0; and aspect ratios of 1:1, 1:2 and 2:1. To reduce redundancy, NMS is used with an IoU threshold of 0.7. In training and testing we use 300 RPN proposals.

Evaluation Metrics: We use the mean average precision (mAP) metric from PASCAL VOC [6], using IoU thresholds of 0.05, 0.5 and 0.75 similar to MS-COCO [8]. For each class, we only report results on $I^{c_n \in C_N}$, these are all images where class c_n has been annotated.

Results: We report results in Table 5 for many-shot classes (those with $\geq$ 100 bounding boxes in training) and few shot classes (with $\geq$ 10 and $<$ 100 bounding boxes in training), alongside AP for the 15 most frequent classes. There are a total of 202 many-shot classes and 78 are few-shot. One can see that our objects are generally harder to detect than in most existing datasets, with performance at the standard IoU$>$ 0.5 below 30%. Even at a very small IoU threshold, the performance is relatively low. The more challenging classes are "meat", "knife", and "spoon", despite being some of the most frequent ones. Notice that the performance for the low-shot regime is substantially lower than in the many-shot regime, falling short of 10%. This points to interesting challenges for the future. However, performances for the *Seen* and *Unseen* splits in object detection are comparable, thus showing generalization capability across environments.

Figure 10 shows qualitative results with detections shown in color and ground truth shown in black. The examples in the right-hand column are failure cases.

4.2 Action Recognition Benchmark

Challenge: Given an action segment $A_i = [t_{s_i}, t_{e_i}]$, we aim to classify the segment into its action class, where classes are defined as $C_a = \{(c_v \in C_V, c_n \in C_N)\}$, and c_n is the first noun in the narration when multiple nouns are present. Note that our dataset supports more complex action-level challenges, such as action localization in the videos of full duration. We decided to focus on the classification challenge first (the segment is provided) since most existing works tackle this challenge. In the future, we aim to provide challenges on action localization, as well as video parsing.
Network Architecture: We train the Temporal Segment Network (TSN) [43] as a state-of-the-art architecture in action recognition, but adjust the output layer to predict both verb and noun classes jointly, with independent losses, as in [44]. We use the PyTorch implementation [45] with the Inception architecture [46], batch normalization [47] and pre-trained on ImageNet [7]. We set the number of temporal segments to 3 in our experiments.
Implementation Details: We train both spatial and temporal streams, the latter on dense optical flow at 30*fps* extracted using the TV-L_1 algorithm [48] between RGB frames using the formulation TV-$L_1(I_{2t}, I_{2t+3})$ to eliminate optical flicker. We will release the computed flow as part of the dataset. We do not perform stratification or weighted sampling, allowing the dataset

TABLE 6: Baseline results for the action recognition challenge

		Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
		VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION
S1	RGB	45.25	35.78	18.91	**86.07**	**62.80**	**39.39**	54.94	40.41	07.01	**23.31**	**30.03**	**05.29**
	FLOW	43.27	17.92	09.10	79.89	39.63	21.91	**64.58**	24.51	01.52	15.35	09.72	01.28
	FUSION	**47.36**	**36.05**	**19.44**	84.27	61.05	35.45	63.12	**44.24**	**07.33**	21.95	29.25	05.22
S2	RGB	35.96	21.74	09.96	**74.70**	44.95	**24.59**	45.40	22.14	02.06	11.79	16.75	**01.91**
	FLOW	**40.56**	14.91	07.28	73.66	33.87	18.29	44.83	22.99	00.92	**14.16**	08.79	00.94
	FUSION	39.67	**22.33**	**10.84**	74.53	**45.23**	23.52	**59.60**	**23.65**	**02.09**	13.37	**16.84**	01.84

TABLE 7: Sample baseline action recognition per-class metrics (using fusion)

		15 Most Frequent Verb Classes														
		put	take	wash	open	close	cut	mix	pour	move	turn-on	remove	turn-off	throw	dry	peel
S1	RECALL	65.32	51.01	80.45	60.98	27.13	74.27	52.63	24.87	00.00	05.63	01.58	03.67	10.11	29.73	26.09
	PRECISION	35.62	41.24	63.17	72.67	72.46	69.38	69.52	66.20	-	53.33	66.67	50.00	56.25	88.00	54.55
S2	RECALL	64.16	48.03	87.76	42.06	15.10	45.69	35.85	06.06	00.00	00.00	00.81	00.00	00.00	00.00	00.00
	PRECISION	30.19	30.46	67.79	57.31	61.54	85.48	65.52	40.00	-	00.00	100.0	-	-	-	00.00

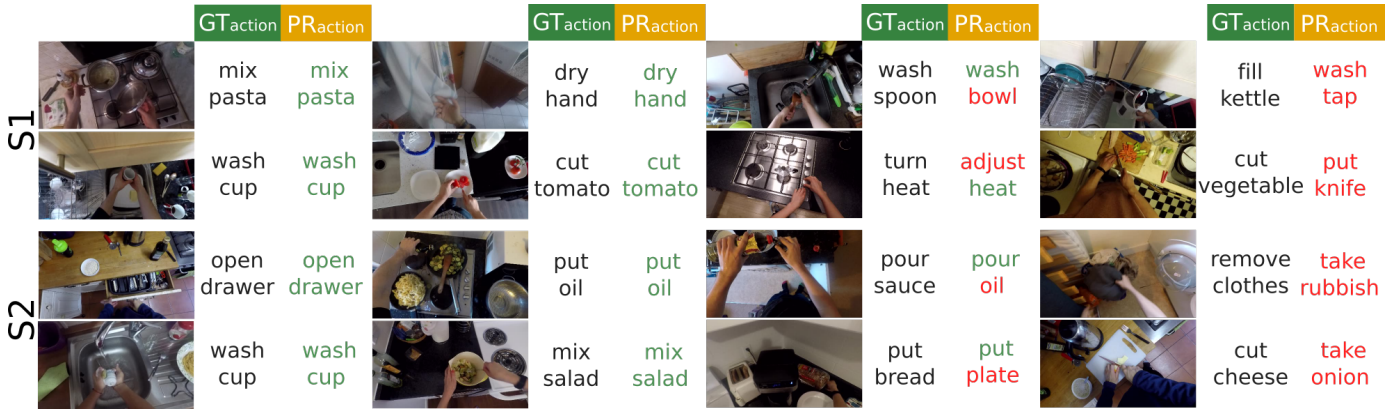

Fig. 11: Qualitative results for the action recognition challenge

class imbalance to propagate into the minibatch. We train each model on 8 Nvidia P100 GPUs on a single compute node (Nvidia DGX-1) for 80 epochs with a minibatch size of 512. We set learning rate to 0.01 for spatial and 0.001 for temporal streams decreasing it by a factor of 10 after epochs 20 and 40. After averaging the 25 samples within the action segment each with 10 spatial croppings as in [43], we fuse both streams by averaging class predictions with equal weights. All unspecified parameters use the same values as [43].

Evaluation Metrics: We report two sets of metrics: aggregate and per-class, which are equivalent to the class-agnostic and class-aware metrics in [12]. For aggregate metrics, we compute top-1 and top-5 accuracy for correct predictions of c_v, c_n and their combination (c_v, c_n) – we refer to these as 'verb', 'noun' and 'action'. Accuracy is reported on the full test set. For per-class metrics, we compute precision and recall, for classes with more than 100 samples in training, then average the metrics across classes - these are 26 verb classes, 70 noun classes. We also report per-class metrics for the valid combinations of these classes - 820 action classes. Per-class metrics for smaller classes are ≈ 0 as TSN is better suited for classes with sufficient training data.

Results: We report results in Table 6 for aggregate metrics and per-class metrics. Fused results perform best or are comparable to the best stream (spatial/temporal). The challenge of getting both verb and noun labels correct remains significant for both *seen* (top-1 accuracy 19.4%) and *unseen* (top-1 accuracy 10.8%) environments. This implies that for many examples, we only get one of the two labels (verb/noun) right. Results also show that generalizing to *unseen* environments is a harder challenge for actions than it is for objects. We give a breakdown per-class metrics for the 15 largest verb classes in Table 7.

Fig. 11 reports qualitative results, with success highlighted in green, and failures in red. In the first two columns, both the verb and the noun are correctly predicted, in the third column one of them is correctly predicted, while in the last column both are incorrect. Challenging cases like distinguishing 'adjust heat' from turning it on, or pouring soy sauce vs oil are shown.

4.3 Action Anticipation Benchmark

Challenge: Anticipating the next action is a well-mastered skill by humans, and automating it has direct implications in assistive living. Given any of the upcoming wearable system (e.g. Microsoft Hololens or Google Glass), anticipating the wearer's next action, from a first-person view, could trigger smart home appliances, providing a seamless achievement of the wearer's goals. Previous works have investigated different anticipation tasks from an egocentric perspective, e.g. predicting future localization [49] or next-active object [50]. We here consider the task of forecasting an action before it happens. Let τ_a be the 'anticipation time', how far in advance to recognize the action, and τ_o be the 'observation time', the length of the observed video segment preceding the action. Given an action segment $A_i = [t_{s_i}, t_{e_i}]$, we predict the action class C_a by observing the video segment *preceding* the action start time t_{s_i} by τ_a, that is $[t_{s_i} - (\tau_a + \tau_o), t_{s_i} - \tau_a]$.

TABLE 8: Baseline results for the action anticipation challenge

		Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
		VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION
S1	RGB	**32.03**	**16.21**	**5.37**	**77.13**	**42.57**	**17.15**	24.62	19.61	**2.28**	**9.46**	**10.72**	**1.56**
	FLOW	30.01	10.75	2.73	74.18	32.00	11.01	19.88	11.32	0.71	7.01	4.27	0.43
	FUSION	31.74	15.06	4.54	76.74	41.50	16.25	**25.10**	**21.59**	1.74	8.32	8.06	0.98
S2	RGB	25.53	9.41	**2.45**	**69.50**	29.36	9.41	8.41	3.91	**0.42**	6.03	**4.06**	**0.41**
	FLOW	26.15	8.55	1.52	68.47	26.36	8.48	11.78	4.32	0.34	**6.65**	3.29	0.30
	FUSION	**26.43**	**9.51**	1.96	69.12	**29.43**	**9.55**	**14.90**	**7.32**	0.31	6.11	3.83	0.37

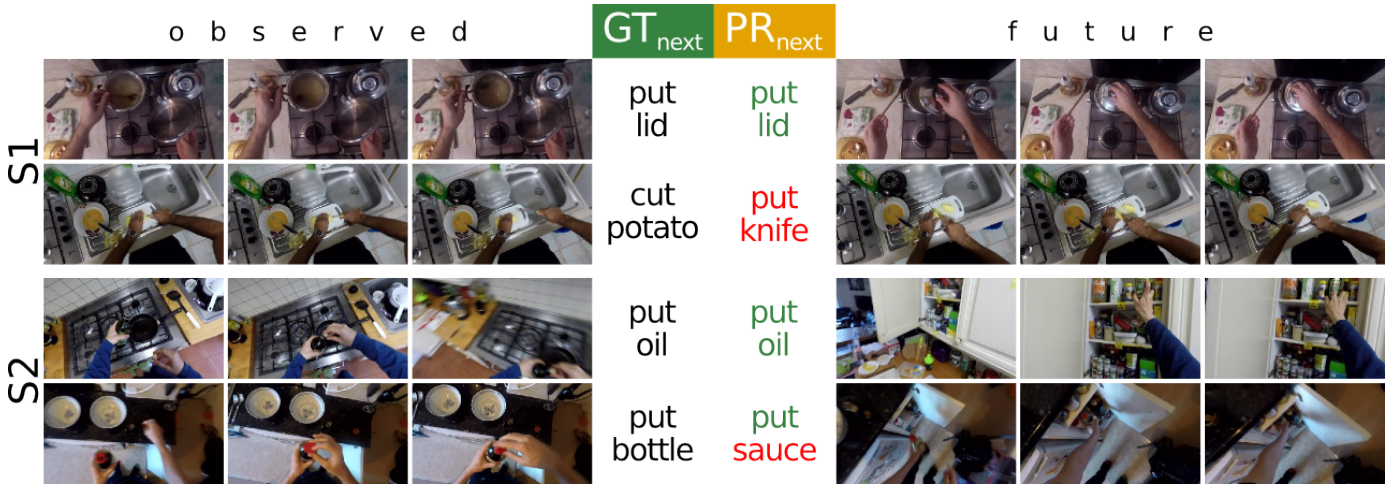

Fig. 12: Qualitative results for the action anticipation challenge

Network Architecture: As in Sec. 4.2, we train TSN [43] to provide baseline action anticipation results. We feed the model with the video segments preceding annotated actions and train it to predict verb and noun classes jointly as in [44]. Similarly to [51], we set $\tau_a = 1s$. We report results with $\tau_o = 1s$, and note that performance drops with longer segments.

Implementation Details: Models for both spatial and temporal modalities are trained using a single Nvidia Titan X with a batch size of 64, for 80 epochs, setting the initial learning rate to 0.001 and dropping it by a factor of 10 after 30 and 60 epochs. Fusion weights spatial and temporal streams with 0.6 and 0.4 respectively. All other parameters use the values specified in [43].

Evaluation Metrics: We use the same evaluation metrics as in Sec. 4.2.

Results: TABLE 8 reports baseline results for the action anticipation challenge. As expected, this is a harder challenge than action recognition, and thus we note a drop in performance throughout. Predicting the object to use next is also more challenging than the action's verb. Fig. 12 reports qualitative results. Success examples are highlighted in green, and failure cases in red. Notice the low per-class precision results for this challenge. As the qualitative figure shows, the method over-predicts 'put' as the next action. Once an object is picked up, the learned model has a tendency to believe it will be put down next. Methods that focus on long-term understanding of the goal, as well as multi-scale history would be needed to circumvent such a tendency.

4.4 Discussion:

The three defined challenges form the base for higher-level understanding of the wearer's goals. We have shown that existing methods are still far from tackling these tasks with high precision, pointing to exciting future directions. Our dataset lends itself naturally to a variety of less explored tasks. We are planning to provide a wider set of challenges, including action localization [52], video parsing [16], and skill determination [53] (e.g., how good are you at making your eggs for breakfast?). Since real-time performance is crucial in this domain, our leaderboard will reflect this, pressing the community to come up with efficient and effective solutions.

5 Conclusion and Future Work

We present the largest and most varied dataset in egocentric vision to date, EPIC-KITCHENS, captured in participants' native environments. We collect 55 hours of video data recorded on a head-mounted Go-Pro, and annotate it with narrations, action segments and object annotations using a pipeline that starts with live commentary of recorded videos by the participants themselves. Baseline results on object detection, action recognition and anticipation challenges show the great potential of the dataset for pushing approaches that target fine-grained video understanding to new frontiers. Dataset, pre-trained models, code and online leaderboard for the three challenges will be released upon publication.

Dataset Release:

- Dataset sequences, extracted frames and optical flow are available at: http://dx.doi.org/10.5523/bris.3h91syskeag572hl6tvuovwv4d
- Annotations, challenge leader-board results and updates and news are available at: http://epic-kitchens.github.io

Acknowledgments

The authors would like to thank all 32 subjects who participated in the dataset collection.

The dataset annotation and release has been sponsored by a charitable donation from Nokia Technologies and the University of Bristol's Jean Golding Institute.
Research at the University of Bristol is supported by EPSRC DTP, EPSRC GLANCE (EP/N013964/1) and EPSRC LOCATE (EP/N033779/1).
Research at the University of Catania is sponsored by Piano della Ricerca 2016-2018 linea di Intervento 2 of DMI.
The object detection benchmark baseline results have been helped by code from, and discussions with, Davide Acuña.

APPENDIX A
SEEN KITCHENS (S1)

P01_11	P01_12	P01_13	P01_14	P01_15	P02_12	P02_13
P02_14	P02_15	P03_21	P03_22	P03_23	P03_24	P03_25
P03_26	P04_24	P04_25	P04_26	P04_27	P04_28	P04_29
P04_30	P04_31	P04_32	P04_33	P05_07	P05_09	P06_10
P06_11	P06_12	P06_13	P06_14	P07_12	P07_13	P07_14
P07_15	P07_16	P07_17	P07_18	P08_09	P08_10	P08_14
P08_15	P08_16	P08_17	P10_03	P12_03	P12_08	P13_01
P13_02	P13_03	P14_06	P14_08	P15_04	P15_05	P15_06
P16_04	P17_02	P19_05	P19_06	P20_05	P20_06	P20_07
P21_02	P22_01	P22_02	P22_03	P22_04	P23_05	P24_09
P25_06	P25_07	P25_08	P26_30	P26_31	P26_32	P26_33
P26_34	P26_35	P26_36	P26_37	P26_38	P26_39	P26_40
P26_41	P27_05	P28_15	P28_16	P28_17	P28_18	P28_19
P28_20	P28_21	P28_22	P28_23	P28_24	P28_25	P28_26
P29_05	P29_06	P30_07	P30_08	P30_09	P31_10	P31_11
P31_12						

APPENDIX B
UNSEEN KITCHENS (S2)

P09_01	P09_02	P09_03	P09_04	P09_05	P09_06
P09_07	P09_08	P11_01	P11_02	P11_03	P11_04
P11_05	P11_06	P11_07	P11_08	P11_09	P11_10
P11_11	P11_12	P11_13	P11_14	P11_15	P11_16
P11_17	P11_18	P11_19	P11_20	P11_21	P11_22
P11_23	P11_24	P18_01	P18_02	P18_03	P18_04
P18_05	P18_06	P18_07	P18_08	P18_09	P18_10
P18_11	P18_12	P32_01	P32_02	P32_03	P32_04
P32_05	P32_06	P32_07	P32_08	P32_09	P32_10

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
[3] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," in *CVPR*, 2015.
[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *ICCV*, 2015.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," in *IJCV*, 2010.
[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.
[9] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017.
[10] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The "something something" video database for learning and evaluating visual common sense," in *ICCV*, 2017.
[11] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark," in *CoRR*, 2016.
[12] H. Zhao, Z. Yan, H. Wang, L. Torresani, and A. Torralba, "SLAC: A Sparsely Labeled Dataset for Action Classification and Localization," *arXiv preprint arXiv:1712.09374*, 2017.
[13] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A Dataset for Movie Description," in *CVPR*, 2015.
[14] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding stories in movies through question-answering," in *CVPR*, 2016.
[15] D. F. Fouhey, W.-c. Kuo, A. A. Efros, and J. Malik, "From lifestyle vlogs to everyday interactions," *arXiv preprint arXiv:1712.02310*, 2017.
[16] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *ECCV*, 2016.
[17] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *ICRA*, 2017.
[18] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *ICRA*, 2018.
[19] Georgia Tech, "Extended GTEA Gaze+," http://webshare.ipat.gatech.edu/coc-rim-wall-lab/web/yli440/egtea_gp, 2018.
[20] A. Fathi, Y. Li, and J. Rehg, "Learning to recognize daily actions using gaze," in *ECCV*, 2012.
[21] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas, "You-do, I-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video." in *BMVC*, 2014.
[22] F. De La Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran, "Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) database," in *Robotics Institute*, 2008.
[23] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *CVPR*, 2012.
[24] H. Kuehne, A. Arslan, and T. Serre, "The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities," in *CVPR*, 2014.
[25] S. Stein and S. McKenna, "Combining Embedded Accelerometers with Computer Vision for Recognizing Food Preparation Activities," in *UbiComp*, 2013.
[26] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A Database for Fine Grained Activity Detection of Cooking Activities," in *CVPR*, 2012.
[27] S. Alletto, G. Serra, S. Calderara, and R. Cucchiara, "Understanding social relationships in egocentric vision," in *Pattern Recognition*, 2015.
[28] A. Fathi, J. Hodgins, and J. Rehg, "Social interactions: A first-person perspective," in *CVPR*, 2012.
[29] M. S. Ryoo and L. Matthies, "First-person activity recognition: What are they doing to me?" in *CVPR*, 2013.
[30] L. Zhou, C. Xu, and J. J. Corso, "Towards automatic learning of procedures from web instructional videos," *arXiv preprint arXiv:1703.09788*, 2017.
[31] J.-B. Alayrac, J. Sivic, I. Laptev, and S. Lacoste-Julien, "Joint discovery of object states and manipulation actions," in *ICCV*, 2017.
[32] Google, "Google cloud speech api," https://cloud.google.com/speech.
[33] IBM, "IBM watson speech to text," https://www.ibm.com/watson/services/speech-to-text.
[34] Carnegie Mellon University, "CMU sphinx," https://cmusphinx.github.io/.
[35] D. Moltisanti, M. Wray, W. Mayol-Cuevas, and D. Damen, "Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video," in *ICCV*, 2017.
[36] Y. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *CVPR*, 2012.
[37] K. Yamaguchi, "Bbox-annotator," https://github.com/kyamagu/bbox-annotator.
[38] G. Miller, "Wordnet: a lexical database for english," in *CACM*, 1995.
[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[40] S. Banerjee and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet," in *CICLing*, 2002.

[41] J. Huang, V. Rathod, D. Chow, C. Sun, M. Zhu, A. Fathi, and Z. Lu, "Tensorflow Object Detection API," https://github.com/tensorflow/models/tree/master/research/object_detection.
[42] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017.
[43] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.
[44] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Joint learning of object and action detectors," in *ICCV*, 2017.
[45] X. Yuanjun, "PyTorch Temporal Segment Network," https://github.com/yjxiong/tsn-pytorch, 2017.
[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.
[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.
[48] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Pattern Recognition*, 2007.
[49] H. S. Park, J.-J. Hwang, Y. Niu, and J. Shi, "Egocentric future localization." in *CVPR*, 2016.
[50] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella, "Next-active-object prediction from egocentric videos," in *JVCIR*, 2017.
[51] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in *CVPR*, 2016.
[52] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *IJCV*, 2018.
[53] H. Doughty, D. Damen, and W. Mayol-Cuevas, "Who's better? who's best? pairwise deep ranking for skill determination," in *CVPR*, 2018.