Scaling Egocentric Vision: The EPIC-KITCHENS Dataset

Dima Damen[1(✉)], Hazel Doughty[1], Giovanni Maria Farinella[2], Sanja Fidler[3], Antonino Furnari[2], Evangelos Kazakos[1], Davide Moltisanti[1], Jonathan Munro[1], Toby Perrett[1], Will Price[1], and Michael Wray[1]

[1] University of Bristol, Bristol, UK
dima.damen@bristol.ac.uk
[2] University of Catania, Catania, Italy
[3] University of Toronto, Toronto, Canada

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 **non-scripted** 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 cooking styles. Our dataset features 55h of video consisting of 11.5M frames, which we densely labelled for a total of 39.6K action segments and 454.3K 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.

Keywords: 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 [19], object detection [37], captioning [26] and visual question-answering [3]. This success has in large part been due to advances in deep learning [27] as well as the availability of large-scale image benchmarks [9,11,30,55]. While gaining attention, work in video understanding has

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01225-0_44) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11208, pp. 753–771, 2018.
https://doi.org/10.1007/978-3-030-01225-0_44

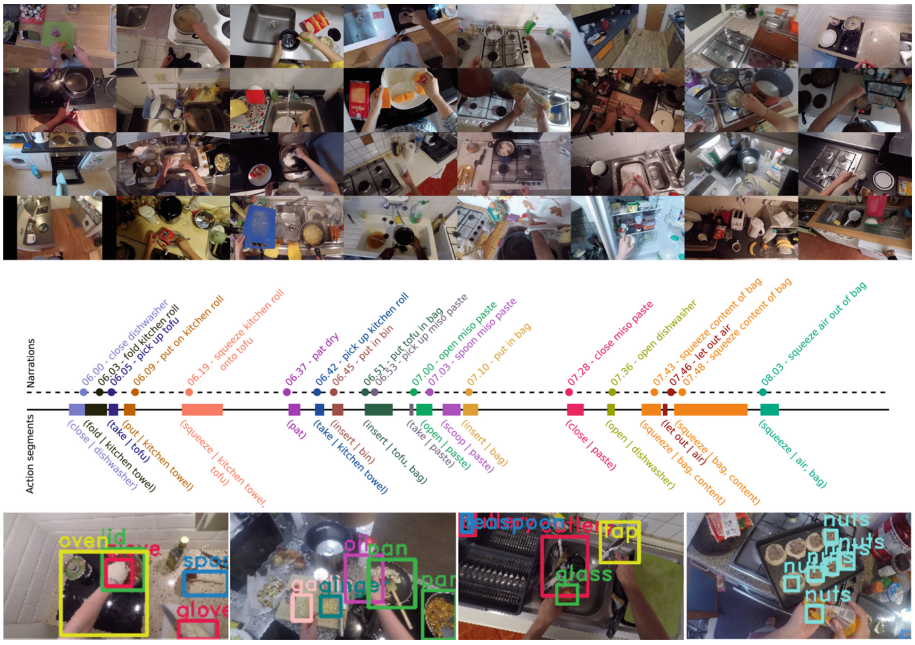

Fig. 1. From top: Frames from the 32 environments; Narrations by participants used to annotate action segments; Active object bounding box annotations

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 [1,14,18,38,46,54]. With the exception of [46], most of these datasets contain videos that are very short in duration, i.e., only a few seconds long, focusing on a single action. Charades [42] makes a step towards activity recognition by collecting 10 K 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 [34,53], and has a direct impact on HCI applications. However, datasets to evaluate first-person vision algorithms [6,8,13,16,36,41] have been significantly smaller in size than their third-person counterparts, often captured in a single environment [6,8,13,16]. 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.

Altogether, EPIC-KITCHENS has 55 h 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. The dataset and leaderboards to track the community's progress on all challenges, with held out test ground-truth are at: http://epic-kitchens.github.io.

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*	454,255	323	32	32
EGTEA Gaze+ [16]	✓	×	×	2018	2.4M	86	10,325	106	0	0	32	1
Charades-ego [41]	70% ✓	×	✓	2018	2.3M	2,751	30,516	157	0	38	71	N/A
BEOID [6]	✓	×	×	2014	0.1M	58	742	34	0	0	5	1
GTEA Gaze+ [13]	✓	×	×	2012	0.4M	35	3,371	42	0	0	13	1
ADL [36]	✓	×	✓	2012	1.0M	20	436	32	137,780	42	20	20
CMU [8]	✓	×	×	2009	0.2M	16	516	31	0	0	16	1
YouCook2 [56]	×	✓	✓	2018	@30fps 15.8M	2,000	13,829	89	0	0	2K	N/A
VLOG [14]	×	✓	✓	2017	37.2M	114K	0	0	0	0	10.7K	N/A
Charades [42]	×	×	✓	2016	7.4M	9,848	67,000	157	0	0	N/A	267
Breakfast [28]	×	✓	✓	2014	3.0M	433	3078	50	0	0	52	18
50 Salads [44]	×	×	×	2013	0.6M	50	2967	52	0	0	25	1
MPII Cooking 2 [39]	×	×	×	2012	2.9M	273	14,105	88	0	0	30	1

2 Related Datasets

We compare EPIC-KITCHENS to four commonly-used [6,8,13,36] and two recent [16,41] egocentric datasets in Table 1, as well as six third-person activity-recognition datasets [14,28,39,42,44,56] that focus on object-interaction activities. We exclude egocentric datasets that focus on inter-person interactions [2,12,40], 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 [14,18,28,41,42]. [28] focuses on cooking dishes based on a list of breakfast recipes. In [14], short segments linked to interactions with 30 daily objects are collected by querying YouTube, while [18,41,42] are scripted – subjects are requested to enact a crowd-sourced storyline [41,42]

or a given action [18], which oftentimes results in less natural looking actions. All 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 [36] 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.

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. 2. Instructions used to collect video narrations from our participants

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, 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 GoPro 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 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 1920×1080, 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 1280×720 and 0.5% at 1920×1440. 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 1.7 h, with the maximum being 4.6 h. Cooking a single meal can span multiple sequences, depending on whether one stays in the kitchen, or leaves and returns later. On average, each participant recorded 13.6 sequences. Figure 3 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. These are analogous to a *live commentary* of the video. The general instructions for narrations are listed in Fig. 2. 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 3 shows wordles of the most frequent words in each language.

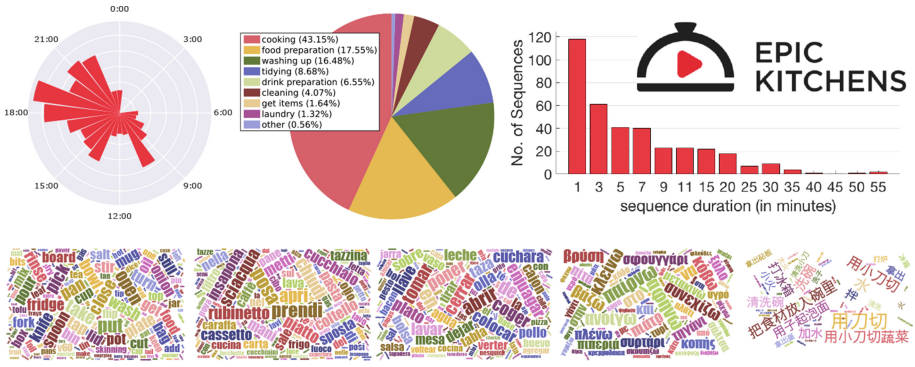

Fig. 3. Top (left to right): time of day of the recording, pie chart of high-level goals, histogram of sequence durations and dataset logo; **Bottom**: Wordles of narrations in native languages (English, Italian, Spanish, Greek and Chinese)

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 labelling.

Table 2. Extracts from 6 transcription files in .sbv format

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

We tested several automatic audio-to-text APIs [5,17,23], 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 s 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.

In total, we collected 39,596 action narrations, corresponding to a narration every 4.9 s in the video. The average number of words per phrase is 2.8 words. These narrations give us an initial labelling of all actions with rough temporal alignment, obtained from 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 labelled 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 Sect. 3.2).
- Participants use their own vocabulary and free language. While this is a challenging issue, we believe it is important to push the community to go beyond the pre-selected list of labels (also argued in [55]). We here resolve this issue by grouping verbs and nouns into minimally overlapping classes (see Sect. 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 localisation, we use a clip from our previous work's understanding [33] 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 s 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. Given one annotation $A_i(j)$ (i is the action and j indexes the annotator), we calculate the agreement as follows: $\alpha_i(j) = \frac{1}{K_a} \sum_{k=1}^{\mathcal{K}_a} \text{IoU}(A_i(j), A_i(k))$. We first find the annotator with the maximum agreement $\hat{j} = \arg\max_j \alpha_i(j)$, and find $\hat{k} = \arg\max_k \text{IoU}(A_i(\hat{j}), A_i(k))$. The ground-truth action segment A_i is then defined as:

$$A_i = \begin{cases} \text{Union}(A_i(\hat{j}), A_i(\hat{k})), & \text{if IoU}(A_i(\hat{j}), A_i(\hat{k})) > 0.5 \\ A_i(\hat{j}), & \text{otherwise} \end{cases} \quad (1)$$

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 4 shows examples of combining annotations.

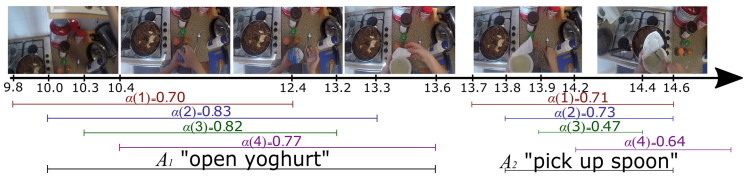

Fig. 4. An example of annotated action segments for 2 consecutive actions

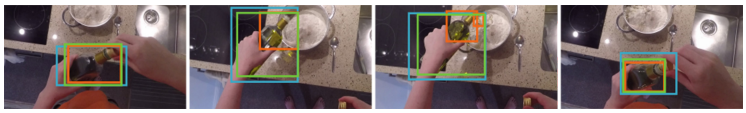

Fig. 5. Object annotation from three AMT workers (orange, blue and green). The green participant's annotations are selected as the final annotations (Color figure online)

In total, we collected such labels for 39,564 action segments (lengths: $\mu = 3.7s$, $\sigma = 5.6$ s). These represent 99.9% of narrated segments. The missed annotations were those labelled 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 [6,29]. 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 [49] for annotating bounding boxes on AMT. Each HIT aims to get an annotation for one object, for the maximum duration of 25 s, which corresponds to 50 consecutive frames at 2 *fps*. 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:

$$\beta(q) = \sum_{f} \max_{j \neq q}^{\mathcal{K}_o} \max_{k,l} \text{IoU}(\text{BB}(j, f, k), \text{BB}(q, f, l)) \tag{2}$$

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 5 shows multiple annotations for four keyframes in a sequence.

Overall, 77% of requested annotations resulted in at least one bounding box. In total, we collected 454,255 bounding boxes ($\mu = 1.64$ boxes/frame, $\sigma = 0.92$). Sample action segments and object bounding boxes are shown in Fig. 6.

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. We 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. We select 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').

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 [32], Word2Vec [31], and Lesk algorithm [4], 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'.

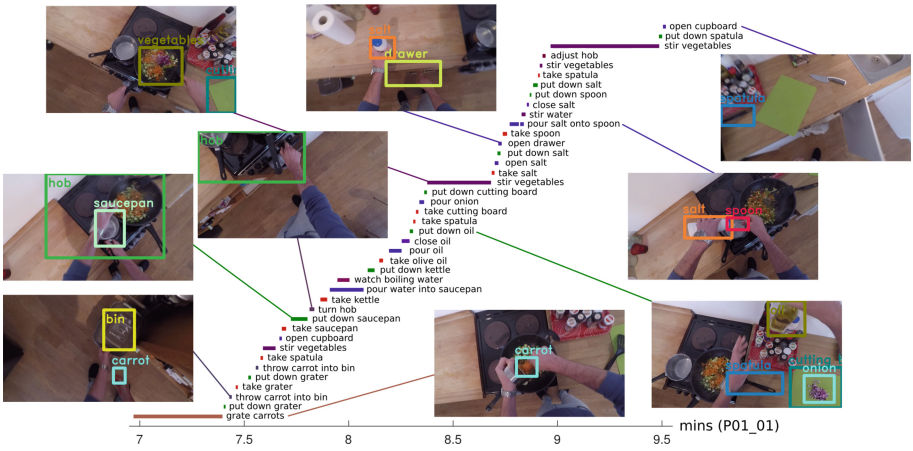

Fig. 6. Sample consecutive action segments with keyframe object annotations

In total, we have 125 C_V classes and 331 C_N classes. Table 3 shows a sample of grouped verbs and nouns into classes. These classes are used in all three defined challenges. In Fig. 7, we show C_V 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. Co-occurring classes are presented in Fig. 8.

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 labelled – 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 [54].

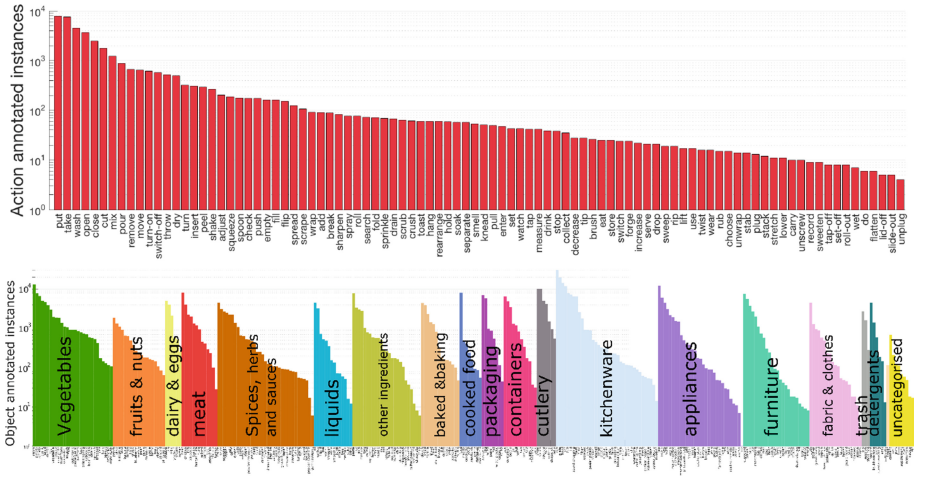

Fig. 7. Top: Frequency of verb classes in action segments; **Bottom**: Frequency of noun clusters in bounding box annotations, by category

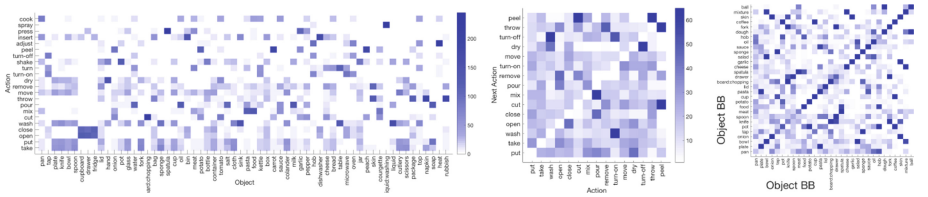

Fig. 8. Left: Frequently co-occurring verb/nouns in action segments [e.g. (open/close, cupboard/drawer/fridge), (peel, carrot/onion/potato/peach), (adjust, heat)]; **Middle**: Next-action excluding repetitive instances of the same action [e.g. peel → cut, turn-on → wash, pour → mix].; **Right**: Co-occurring bounding boxes in one frame [e.g. (pot, coffee), (knife, chopping board), (tap, sponge)]

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, and avail online leaderboards. 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.

Table 3. Sample verb and noun classes

	ClassNo (Key)	Clustered Words
VERB	0 (take)	Take, grab, pick, get, fetch, pick-up, ...
	3 (close)	Close, close-off, shut
	12 (turn-on)	Turn-on, start, begin, ignite, switch-on, activate, restart, light, ...
NOUN	1 (pan)	Pan, frying pan, saucepan, wok, ...
	8 (cupboard)	Cupboard, cabinet, locker, flap, cabinet door, cupboard door, closet, ...
	51 (cheese)	Cheese slice, mozzarella, paneer, parmesan, ...
	78 (top)	Top, counter, counter top, surface, kitchen counter, kitchen top, tiles, ...

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		28,587	28,561	326,388
S1 Test	28	106	39084	20%	8,069	8,064	97,872
S2 Test	4	54	13231	7%	2,939	2,939	29,995

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 analyse the capabilities of the approaches to quickly learn novel objects (with only a few examples). Our challenge leaderboard reflects the methods' abilities on both sets of classes.

Method: We evaluate object detection using Faster R-CNN [37] 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 [21, 22] with a base architecture of ResNet-101 [19] pre-trained on MS-COCO [30].

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	78.40	74.34	66.86	65.40	86.40	68.32	49.96	45.79	39.59	48.31	58.59	61.85	77.65	52.17	62.46	31.59	51.60	47.84
	IoU > 0.5	70.63	68.21	61.93	41.92	73.04	62.90	33.77	26.96	27.69	38.10	50.07	51.71	69.74	36.00	58.64	20.72	38.81	35.41
	IoU > 0.75	22.26	46.34	36.98	3.50	26.59	20.47	4.13	2.48	5.53	9.39	13.21	11.25	22.61	7.37	30.53	2.70	10.07	8.69
S2	IoU > 0.05	80.35	88.38	66.79	47.65	83.40	71.17	63.24	46.36	71.87	29.91	N/A	55.36	78.02	55.17	61.55	23.19	49.30	46.64
	IoU > 0.5	67.42	85.62	62.75	26.27	65.90	59.22	44.14	30.30	56.28	24.31	N/A	47.00	73.82	39.49	51.56	16.95	34.95	33.11
	IoU > 0.75	18.41	60.43	33.32	2.21	6.41	14.55	4.65	1.77	12.80	7.40	N/A	7.54	36.94	9.45	22.1	2.46	8.68	8.05

Fig. 9. Qualitative results for the object detection challenge

Implementation Details: Learning rate is initialised to 0.0003 decaying by a factor of 10 after 90 K and stopped after 120 K iterations. We use a mini-batch 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 mini-batch size of 32. As in [37], 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: 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. We use the mean average precision (mAP) metric from PASCAL VOC [11], using IoU thresholds of 0.05, 0.5 and 0.75 similar to [30].

Results: We report results in Table 5 for many-shot classes (those with ≥ 100 bounding boxes in training) and few shot classes (with ≥ 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 88 few-shot classes. 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 40%. 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. 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 9 shows qualitative results with detections shown in colour 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 localisation 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.

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	Chance/Random	12.62	1.73	00.22	43.39	08.12	03.68	03.67	01.15	00.08	03.67	01.15	00.05
	Largest Class	22.41	04.50	01.59	70.20	18.89	14.90	00.86	00.06	00.00	03.84	01.40	00.12
	2SCNN (FUSION)	42.16	29.14	13.23	80.58	53.70	30.36	29.39	30.73	5.35	14.83	21.10	04.46
	TSN (RGB)	45.68	**36.80**	19.86	**85.56**	**64.19**	**41.89**	**61.64**	34.32	09.96	**23.81**	**31.62**	08.81
	TSN (FLOW)	42.75	17.40	09.02	79.52	39.43	21.92	21.42	13.75	02.33	15.58	09.51	02.06
	TSN (FUSION)	**48.23**	36.71	**20.54**	84.09	62.32	39.79	47.26	**35.42**	**10.46**	22.33	30.53	**08.83**
S2	Chance/Random	10.71	01.89	00.22	38.98	09.31	03.81	03.56	01.08	00.08	03.56	01.08	00.05
	Largest Class	22.26	04.80	00.10	63.76	19.44	17.17	00.85	00.06	00.00	03.84	01.40	00.12
	2SCNN (FUSION)	36.16	18.03	07.31	71.97	38.41	19.49	18.11	15.31	02.86	10.52	12.55	02.69
	TSN (RGB)	34.89	21.82	10.11	**74.56**	45.34	**25.33**	19.48	14.67	04.77	11.22	17.24	05.67
	TSN (FLOW)	**40.08**	14.51	06.73	73.40	33.77	18.64	19.98	09.48	02.08	**13.81**	08.58	02.27
	TSN (FUSION)	39.40	**22.70**	**10.89**	74.29	**45.72**	25.26	**22.54**	**15.33**	**05.60**	13.06	**17.52**	**05.81**

Table 7. Sample baseline action recognition per-class metrics (using TSN fusion)

		15 Most Frequent (in Train Set) Verb Classes														
		put	take	wash	open	close	cut	mix	pour	move	turn-on	remove	turn-off	throw	dry	peel
S1	RECALL	67.51	48.27	83.19	63.32	25.45	77.64	50.20	26.32	00.00	08.28	05.11	05.45	24.18	36.49	30.43
	PRECISION	36.29	43.21	63.01	69.74	75.50	68.71	68.51	60.98	-	46.15	53.85	66.67	75.86	81.82	51.85
S2	RECALL	74.23	34.05	83.67	43.64	18.40	33.90	35.85	13.13	00.00	00.00	00.00	00.00	00.00	2.70	00.00
	PRECISION	29.60	30.68	67.06	56.28	66.67	88.89	70.37	76.47	-	-	00.00	-	-	100.0	00.00

Network Architecture: We train the Temporal Segment Network (TSN) [48] 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 [25]. We use the PyTorch implementation [51] with the Inception architecture [45], batch normalization [24] and pre-trained on ImageNet [9].

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 [52] between RGB frames using the formulation TV-$L_1(I_{2t}, I_{2t+3})$ to eliminate optical flicker, and released the computed flow as part of the dataset. We do not perform stratification or weighted sampling, allowing the dataset class imbalance to propagate into the mini-batch. We train each model on 8 Nvidia P100 GPUs on a single compute node (Nvidia DGX-1) for 80 epochs with a mini-batch 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 [48], we fuse both streams by averaging class predictions with equal weights. All unspecified parameters use the same values as [48].

Evaluation Metrics: We report two sets of metrics: aggregate and per-class, which are equivalent to the class-agnostic and class-aware metrics in [54]. 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, 71 noun classes, and 819 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. We compare TSN (3 segments) to 2SCNN [43] (1 segment), chance and largest class baselines. 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 20.5%) and *unseen* (top-1 accuracy 10.9%) environments. This implies that for many examples, we only get one of the two labels (verb/noun) right. Results also show that generalising 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. 10. Qualitative results for the action recognition and anticipation challenges

Figure 10 reports qualitative results, with success highlighted in green, and failures in red. In the first column both the verb and the noun are correctly predicted, in the second column one of them is correctly predicted, while in the third 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 assertive 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 localisation [35] or next-active object [15]. We here consider the task of forecasting an action before it happens. Let τ_a be the 'anticipation time', how far in advance to recognise 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 $[t_{s_i} - (\tau_a + \tau_o), t_{s_i} - \tau_a]$.

Network Architecture: As in Sect. 4.2, we train TSN [48] to provide baseline action anticipation results and compare with 2SCNN [43]. We feed the model with the video segments preceding annotated actions and train it to predict verb and noun classes jointly as in [25]. Similarly to [47], we set $\tau_a = 1\,\text{s}$. We report results with $\tau_o = 1\,\text{s}$, 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 [48].

Evaluation Metrics: We use the same evaluation metrics as in Sect. 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. Unlike the case of action recognition, the flow stream and fusion do not generally improve performances. TSN often offers small, but consistent improvements over 2SCNN.

Figure 10 reports qualitative results. Success examples are highlighted in green, and failure cases in red. 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.

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	2SCNN (RGB)	29.76	15.15	04.32	76.03	38.56	15.21	13.76	17.19	02.48	07.32	10.72	01.81
	TSN (RGB)	**31.81**	**16.22**	**06.00**	**76.56**	**42.15**	**18.21**	**23.91**	19.13	**03.13**	**09.33**	**11.93**	**02.39**
	TSN (FLOW)	29.64	10.30	02.93	73.70	30.09	10.92	18.34	10.70	01.41	06.99	05.48	01.00
	TSN (FUSION)	30.66	14.86	04.62	75.32	40.11	16.01	08.84	**21.85**	02.25	06.76	09.15	01.55
S2	2SCNN (RGB)	25.23	09.97	02.29	**68.66**	27.38	09.35	16.37	06.98	00.85	05.80	06.37	**01.14**
	TSN (RGB)	25.30	**10.41**	**02.39**	68.32	**29.50**	**09.63**	07.63	**08.79**	00.80	06.06	**06.74**	01.07
	TSN (FLOW)	**25.61**	08.40	01.78	67.57	24.62	08.19	10.80	04.99	01.02	**06.34**	04.72	00.84
	TSN (FUSION)	25.37	09.76	01.74	68.25	27.24	09.05	**13.03**	05.13	00.90	05.65	05.58	00.79

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 localisation [50], video parsing [42], visual dialogue [7], goal completion [20] and skill determination [10] (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 GoPro, 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 and online leaderboard for the three challenges are available from http://epic-kitchens.github.io.

Acknowledgement. Annotations sponsored by a charitable donation from Nokia Technologies and UoB's Jean Golding Institute. Research supported by EPSRC DTP, EPSRC GLANCE (EP/N013964/1), EPSRC LOCATE (EP/N033779/1) and Piano della Ricerca 2016–2018 linea di Intervento 2 of DMI. The object detection baseline helped by code from, and discussions with, Davide Acuña.

References

1. Abu-El-Haija, S., et al.: YouTube-8M: a large-scale video classification benchmark. In: CoRR (2016)
2. Alletto, S., Serra, G., Calderara, S., Cucchiara, R.: Understanding social relationships in egocentric vision. Pattern Recogn. (2015)
3. Antol, S., et al.: VQA: visual question answering. In: ICCV (2015)

4. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: CICLing (2002)
5. Carnegie Mellon University: CMU sphinx. https://cmusphinx.github.io/
6. Damen, D., Leelasawassuk, T., Haines, O., Calway, A., Mayol-Cuevas, W.: You-do, I-learn: discovering task relevant objects and their modes of interaction from multi-user egocentric video. In: BMVC (2014)
7. Das, A., et al.: Visual Dialog. In: CVPR (2017)
8. De La Torre, F., et al.: Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database. In: Robotics Institute (2008)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
10. Doughty, H., Damen, D., Mayol-Cuevas, W.: Who's better? who's best? Pairwise deep ranking for skill determination. In: CVPR (2018)
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. IJCV (2010)
12. Fathi, A., Hodgins, J., Rehg, J.: Social interactions: a first-person perspective. In: CVPR (2012)
13. Fathi, A., Li, Y., Rehg, J.M.: Learning to recognize daily actions using gaze. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 314–327. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_23
14. Fouhey, D.F., Kuo, W.c., Efros, A.A., Malik, J.: From lifestyle vlogs to everyday interactions. arXiv preprint arXiv:1712.02310 (2017)
15. Furnari, A., Battiato, S., Grauman, K., Farinella, G.M.: Next-active-object prediction from egocentric videos. JVCIR (2017)
16. Georgia Tech: Extended GTEA Gaze+ (2018). http://webshare.ipat.gatech.edu/coc-rim-wall-lab/web/yli440/egtea_gp
17. Google: Google cloud speech api. https://cloud.google.com/speech
18. Goyal, R., et al.: The "something something" video database for learning and evaluating visual common sense. In: ICCV (2017)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
20. Heidarivincheh, F., Mirmehdi, M., Damen, D.: Action completion: a temporal model for moment detection. In: BMVC (2018)
21. Huang, J., et al.: Tensorflow Object Detection API. https://github.com/tensorflow/models/tree/master/research/object_detection
22. Huang, J., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR (2017)
23. IBM: IBM watson speech to text. https://www.ibm.com/watson/services/speech-to-text
24. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
25. Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C.: Joint learning of object and action detectors. In: ICCV (2017)
26. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
28. Kuehne, H., Arslan, A., Serre, T.: The language of actions: recovering the syntax and semantics of goal-directed human activities. In: CVPR (2014)

29. Lee, Y., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: CVPR (2012)
30. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
31. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
32. Miller, G.: WordNet: a lexical database for English. In: CACM (1995)
33. Moltisanti, D., Wray, M., Mayol-Cuevas, W., Damen, D.: Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In: ICCV (2017)
34. Nair, A., et al.: Combining self-supervised learning and imitation for vision-based rope manipulation. In: ICRA (2017)
35. Park, H.S., Hwang, J.J., Niu, Y., Shi, J.: Egocentric future localization. In: CVPR (2016)
36. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: CVPR (2012)
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
38. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR (2015)
39. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: CVPR (2012)
40. Ryoo, M.S., Matthies, L.: First-person activity recognition: what are they doing to me? In: CVPR (2013)
41. Sigurdsson, G.A., Gupta, A., Schmid, C., Farhadi, A., Alahari, K.: Charades-ego: a large-scale dataset of paired third and first person videos. In: ArXiv (2018)
42. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: crowdsourcing data collection for activity understanding. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 510–526. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_31
43. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
44. Stein, S., McKenna, S.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: UbiComp (2013)
45. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
46. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: MovieQA: understanding stories in movies through question-answering. In: CVPR (2016)
47. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: CVPR (2016)
48. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
49. Yamaguchi, K.: Bbox-annotator. https://github.com/kyamagu/bbox-annotator
50. Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: dense detailed labeling of actions in complex videos. IJCV (2018)
51. Yuanjun, X.: PyTorch Temporal Segment Network (2017). https://github.com/yjxiong/tsn-pytorch

52. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L1 optical flow. Pattern Recogn. (2007)
53. Zhang, T., McCarthy, Z., Jow, O., Lee, D., Goldberg, K., Abbeel, P.: Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In: ICRA (2018)
54. Zhao, H., Yan, Z., Wang, H., Torresani, L., Torralba, A.: SLAC: a sparsely labeled dataset for action classification and localization. arXiv preprint arXiv:1712.09374 (2017)
55. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE20K dataset. In: CVPR (2017)
56. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. arXiv preprint arXiv:1703.09788 (2017)