EPIC KITCHENS-55 Dataset

EPIC-KITCHENS-55 is the largest dataset in first-person (egocentric) vision; 55 hours of multi-faceted, non-scripted recordings in native environments - i.e. the wearers' homes, capturing all daily activities in the kitchen over multiple days. Annotations are collected using a novel 'live' audio commentary approach.

Authors

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

- (1 University of Bristol)
- (2 University of Toronto)
- (3 University of Catania)

Contact: uob-epic-kitchens@bristol.ac.uk

Citing

When using the dataset, kindly reference:

```
@INPROCEEDINGS{Damen2018EPICKITCHENS,
   title={Scaling Egocentric Vision: The EPIC-KITCHENS Dataset},
   author={Damen, Dima and Doughty, Hazel and Farinella, Giovanni Maria  and Fidler, Sanja and
           Furnari, Antonino and Kazakos, Evangelos and Moltisanti, Davide and Munro, Jonathan
           and Perrett, Toby and Price, Will and Wray, Michael},
   booktitle={European Conference on Computer Vision (ECCV)},
   year={2018}
}
```

(Check publication here)

Dataset Details

Ground Truth

We provide ground truth for action segments and object bounding boxes.

- **Objects:** Full bounding boxes of narrated objects for every annotated frame.
- **Actions:** Split into narrations and action labels:
  - Narrations containing the narrated sentence with the timestamp.
  - Action labels containing the verb and noun labels along with the start and end times of the segment.

Dataset Splits

The dataset is comprised of three splits with the corresponding ground truth:

- Training set - Full ground truth.

- Seen Kitchens (S1) Test set - Start/end times only.
- Unseen Kitchens (S2) Test set - Start/end times only.

Initially we are only releasing the full ground truth for the training set in order to run action and object challenges.

Important Files

- `README.md (this file)`
- `README.html`
- `README.pdf`
- `license.txt`
- `EPIC_train_action_labels.csv` (Info) (Pickle)
- `EPIC_train_object_labels.csv` (Info)
- `EPIC_test_s1_timestamps.csv` (Info) (Pickle)
- `EPIC_test_s2_timestamps.csv` (Info) (Pickle)
- `EPIC_train_object_action_correspondence.csv` (Info) (Pickle)
- `EPIC_test_s1_object_action_correspondence.csv` (Info) (Pickle)
- `EPIC_test_s2_object_action_correspondence.csv` (Info) (Pickle)
- `EPIC_test_s1_object_video_list.csv` (Info)
- `EPIC_test_s2_object_video_list.csv` (Info)
- `EPIC_noun_classes.csv` (Info)
- `EPIC_verb_classes.csv` (Info)

Additional Files

- `EPIC_train_invalid_labels.csv` (Info) (Pickle)
- `EPIC_train_action_narrations.csv` (Info)
- `EPIC_descriptions.csv` (Info)
- `EPIC_many_shot_verbs.csv` (Info)
- `EPIC_many_shot_nouns.csv` (Info)
- `EPIC_many_shot_actions.csv` (Info)
- `EPIC_video_info.csv` (info)

We direct the reader to RDSF for the videos and rgb/flow frames.

We provide html and pdf alternatives to this README which are auto-generated.

Files Structure

EPIC_train_action_labels.csv

CSV file containing 14 columns:

Column Name	Type	Example	Description
`uid`	int	`6374`	Unique ID of the segment.
`video_id`	string	`P03_01`	Video the segment is in.

Column Name	Type	Example	Description
narration	string	close fridge	English description of the action provided by the participant.
start_timestamp	string	00:23:43.847	Start time in HH:mm:ss.SSS of the action.
stop_timestamp	string	00:23:47.212	End time in HH:mm:ss.SSS of the action.
start_frame	int	85430	Start frame of the action (WARNING only for frames extracted as detailed in Video Information).
stop_frame	int	85643	End frame of the action (WARNING only for frames extracted as detailed in Video Information).
participant_id	string	P03	ID of the participant.
verb	string	close	Parsed verb from the narration.
noun	string	fridge	First parsed noun from the narration.
verb_class	int	3	Numeric ID of the parsed verb's class.
noun_class	int	10	Numeric ID of the parsed noun's class.
all_nouns	list of string (1 or more)	['fridge']	List of all parsed nouns from the narration.
all_noun_classes	list of int (1 or more)	[10]	List of numeric IDs corresponding to all of the parsed nouns' classes from the narration.

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_train_invalid_labels.csv

CSV file containing 14 columns:

Column Name	Type	Example	Description
uid	int	6374	Unique ID of the segment.
video_id	string	P03_01	Video the segment is in.
narration	string	close fridge	English description of the action provided by the participant.
start_timestamp	string	00:23:43.847	Start time in HH:mm:ss.SSS of the action.
stop_timestamp	string	00:23:47.212	End time in HH:mm:ss.SSS of the action.
start_frame	int	85430	Start frame of the action (WARNING only for frames extracted as detailed in Video Information).
stop_frame	int	85643	End frame of the action (WARNING only for frames extracted as detailed in Video Information).
participant_id	string	P03	ID of the participant.
verb	string	close	Parsed verb from the narration.
noun	string	fridge	First parsed noun from the narration.
verb_class	int	3	Numeric ID of the parsed verb's class.
noun_class	int	10	Numeric ID of the parsed noun's class.
all_nouns	list of string (1 or more)	['fridge']	List of all parsed nouns from the narration.

Column Name	Type	Example	Description
all_noun_classes	list of int (1 or more)	[10]	List of numeric IDs corresponding to all of the parsed nouns' classes from the narration.

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_train_action_narrations.csv

CSV file containing 5 columns:

Note: The start/end timestamp refers to the start/end time of the narration, not the action itself.

Column Name	Type	Example	Description
participant_id	string	P03	ID of the participant.
video_id	string	P03_01	Video the segment is in.
start_timestamp	string	00:23:43.847	Start time in HH:mm:ss.SSS of the narration.
stop_timestamp	string	00:23:47.212	End time in HH:mm:ss.SSS of the narration.
narration	string	close fridge	English description of the action provided by the participant.

EPIC_train_object_labels.csv

CSV file containing 6 columns:

Column Name	Type	Example	Description
noun_class	int	20	Integer value representing the class in noun-classes.csv.
noun	string	bag	Original string name for the object.
participant_id	string	P01	ID of participant.
video_id	string	P01_01	Video the object was annotated in.
frame	int	056581	Frame number of the annotated object.
bounding_boxes	list of 4-tuple (0 or more)	"[(76, 1260, 462, 186)]"	Annotated boxes with format (<top:int>,<left:int>,<height:int>,<width:int>).

EPIC_train_object_action_correspondence.csv

CSV file containing 5 columns:

Column Name	Type	Example	Description
participant_id	string	P01	ID of participant.
video_id	string	P01_01	Video the frames are part of.
object_frame	int	56581	Frame number of the object detection image from object_detection_images.

Column Name	Type	Example	Description
action_frame	int	56638	Frame number of the corresponding image in the released frames for action recognition in frames_rgb_flow.
timestamp	string	00:00:00.00	Timestamp in HH:mm:ss.SS corresponding to the frame.

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_test_s1_object_action_correspondence.csv

CSV file containing 5 columns:

Column Name	Type	Example	Description
participant_id	string	P01	ID of participant.
video_id	string	P01_11	Video containing the object s1 test frames.
object_frame	int	33601	Frame number of the object detection image from object_detection_images.
action_frame	int	33635	Frame number of the corresponding image in the released frames for action recognition in frames_rgb_flow.
timestamp	string	00:09:20.58	Timestamp in HH:mm:ss.SS corresponding to the frames.

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_test_s2_object_action_correspondence.csv

CSV file containing 5 columns:

Column Name	Type	Example	Description
participant_id	string	P09	ID of participant.
video_id	string	P09_05	Video containing the object s2 test frames.
object_frame	int	15991	Frame number of the object detection image from object_detection_images.
action_frame	int	16007	Frame number of the corresponding image in the released frames for action recognition in frames_rgb_flow.
timestamp	string	00:04:26.78	Timestamp in HH:mm:ss.SS corresponding to the frames.

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_test_s1_object_video_list.csv

CSV file listing the videos used to obtain the object s1 test frames. The frames can be obtained from RDSF under `object_detection_images/test`. Please test all frames from this folder for the videos listed in this csv.

Column Name	Type	Example	Description
`video_id`	string	`P01_11`	Video containing the object s1 test frames.
`participant_id`	string	`P01`	ID of the participant.

EPIC_test_s2_object_video_list.csv

CSV file listing the videos used to obtain the object s2 test frames. The frames can be obtained from RDSF under `object_detection_images/test`. Please test all frames from this folder for the videos listed in this csv.

Column Name	Type	Example	Description
`video_id`	string	`P01_11`	Video containing the object s2 test frames.
`participant_id`	string	`P01`	ID of the participant.

EPIC_test_s1_timestamps.csv

CSV file containing 7 columns:

Column Name	Type	Example	Description
`uid`	int	`1924`	Unique ID of the segment.
`participant_id`	string	`P01`	ID of the participant.
`video_id`	string	`P01_11`	Video the segment is in.
`start_timestamp`	string	`00:00:00.000`	Start time in `HH:mm:ss.SSS` of the action.
`stop_timestamp`	string	`00:00:01.890`	End time in `HH:mm:ss.SSS` of the action.
`start_frame`	int	`1`	Start frame of the action (WARNING only for frames extracted as detailed in Video Information).
`stop_frame`	int	`93`	End frame of the action (WARNING only for frames extracted as detailed in Video Information).

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_test_s2_timestamps.csv

CSV file containing 7 columns:

Column Name	Type	Example	Description
`uid`	int	`15582`	Unique ID of the segment.
`participant_id`	string	`P09`	ID of the participant.

Column Name	Type	Example	Description
video_id	string	P09_01	Video the segment is in.
start_timestamp	string	00:00:01.970	Start time in HH:mm:ss.SSS of the action.
stop_timestamp	string	00:00:03.090	End time in HH:mm:ss.SSS of the action.
start_frame	int	118	Start frame of the action (WARNING only for frames extracted as detailed in Video Information).
stop_frame	int	185	End frame of the action (WARNING only for frames extracted as detailed in Video Information).

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above. This pickle file was created with pickle protocol 2 on pandas version 0.22.0.

EPIC_noun_classes.csv

CSV file containing 3 columns:

Note: a colon represents a compound noun with the more generic noun first. So pan:dust should be read as dust pan.

Column Name	Type	Example	Description
noun_id	int	2	ID of the noun class.
class_key	string	pan:dust	Key of the noun class.
nouns	list of string (1 or more)	"['pan:dust', 'dustpan']"	All nouns within the class (includes the key).

EPIC_verb_classes.csv

CSV file containing 3 columns:

Column Name	Type	Example	Description
verb_id	int	3	ID of the verb class.
class_key	string	close	Key of the verb class.
verbs	list of string (1 or more)	"['close', 'close-off', 'shut']"	All verbs within the class (includes the key).

EPIC_descriptions.csv

CSV file containing 4 columns:

Column Name	Type	Example	Description
video_id	string	P01_01	ID of the video.
date	string	30/04/2017	Date on which the video was shot.
time	string	13:49:00	Local recording time of the video.
description	string	prepared breakfast with soy milk and cereals	Description of the activities contained in the video.

EPIC_many_shot_verbs.csv

CSV file containing the many shot verbs. A verb class is considered many shot if it appears more than 100 times in training. (NOTE: this file is derived from `EPIC_train_action_labels.csv`, checkout the accompanying notebook demonstrating how we compute these classes)

Column Name	Type	Example	Description
`verb_class`	int	`1`	Numeric ID of the verb class
`verb`	string	`put`	Verb corresponding to the verb class

EPIC_many_shot_nouns.csv

CSV file containing the many shot nouns. A noun class is considered many shot if it appears more than 100 times in training. (NOTE: this file is derived from `EPIC_train_action_labels.csv`, checkout the accompanying notebook demonstrating how we compute these classes)

Column Name	Type	Example	Description
`noun_class`	int	`3`	Numeric ID of the noun class
`noun`	string	`tap`	Noun corresponding to the noun class

EPIC_many_shot_actions.csv

CSV file containing the many shot actions. An action class (composed of a verb class and noun class) is considered many shot if BOTH the verb class and noun class are many shot AND the action class appears in training at least once. (NOTE: this file is derived from `EPIC_train_action_labels.csv`, checkout the accompanying notebook demonstrating how we compute these classes)

Column Name	Type	Example	Description
`action_class`	(int, int)	`(9, 84)`	Numeric Pair of IDs, first the verb, then the noun
`verb_class`	int	`9`	Numeric ID of the verb class
`verb`	string	`move`	Verb corresponding to the verb class
`noun_class`	int	`84`	Numeric ID of the noun class
`noun`	string	`sausage`	Noun corresponding to the noun class

EPIC_video_info.csv

CSV file containing information for each video

Column Name	Type	Example	Description
`video`	(string)	`P01_01`	Video ID
`resolution`	(string)	`1920x1080`	Resolution of the video, format is `WIDTHxHEIGHT`
`duration`	(float)	`1652.152817`	Duration of the video, in seconds
`fps`	(float)	`59.9400599400599`	Frame rate of the video

File Downloads

Due to the size of the dataset we provide scripts for downloading parts of the dataset:

- videos (750GB)
- frames (320GB)
  - rgb-frames (220GB)
  - flow-frames (100GB)
- object annotation images (80GB)

Note: These scripts will work for Linux and Mac. For Windows users a bash installation should work.

These scripts replicate the folder structure of the dataset release, found here.

If you wish to download part of the dataset instructions can be found here.

Video Information

Videos are recorded in 1080p at 59.94 FPS on a GoPro Hero 5 with linear field of view. There are a minority of videos which were shot at different resolutions, field of views, or FPS due to participant error or camera. These videos identified using `ffprobe` are:

- 1280x720: `P12_01`, `P12_02`, `P12_03`, `P12_04`.
- 2560x1440: `P12_05`, `P12_06`
- 29.97 FPS: `P09_07`, `P09_08`, `P10_01`, `P10_04`, `P11_01`, `P18_02`, `P18_03`
- 48 FPS: `P17_01`, `P17_02`, `P17_03`, `P17_04`
- 90 FPS: `P18_09`

The GoPro Hero 5 was also set to drop the framerate in low light conditions to preserve exposure leading to variable FPS in some videos. If you wish to extract frames we suggest you resample at 60 FPS to mitigate issues with variable FPS, this can be achieved in a single step with FFmpeg:

```
ffmpeg -i "P##_**.MP4" -vf "scale=-2:256" -q:v 4 -r 60 "P##_**/frame_%010d.jpg"
```

where `##` is the Participant ID and `**` is the video ID.

Optical flow was extracted using a fork of `gpu_flow` made available on github. We set the parameters: stride = 2, dilation = 3, bound = 25 and size = 256.

License

Changelog

See release history for changelog.