EPIC Kitchens Dataset

Initial Release, April 2018

Website

Authors

Dima Damen (1) Hazel Doughty (1) Sanja Fidler (2) Antonino Furnari (3) Evangelos Kazakos (1) Giovanni Maria Farinella (3) 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-kitchens2018@bristol.ac.uk

Citing

When using the dataset, kindly reference:

> BibTex of Arxiv dataset paper

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 - Timestamp and start/end times only.
- Unseen Kitchens (S2) Test set - Timestamp and 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_narrations.csv` (Info).
- `EPIC_train_action_labels.csv` (Info).
- `EPIC_train_action_labels.pkl` (Info).
- `EPIC_train_object_labels.csv` (Info).
- `EPIC_test_s1_timestamps.csv` (Info).
- `EPIC_test_s2_timestamps.csv` (Info).
- `EPIC_verb_classes.csv` (Info).
- `EPIC_noun_classes.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_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.

Column Name	Type	Example	Description
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_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.
narration	string	'close fridge'	English description of the action provided
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
stop_frame	int	85643	End frame of the action (WARNING only
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_nouns_class	list of int (1 or more)	[10]	List of numeric IDs corresponding to all of

Please note we have included a python pickle file for ease of use. This includes a pandas dataframe with the same layout as above.

EPIC_train_object_labels.csv

CSV file containing 6 columns:

Column Name	Type	Example	Description
noun_class	int	20	Integer value representing the
noun	string	'bag'	Original string name for the o
participant_id	string	'P01'	ID of participant.
video_id	string	'P01_01'	Video the object was annotate
frame	int	056581	Frame number of the annotate
bounding_boxes	list of 4-tuple (0 or more)	"[(76, 1260, 462, 186)]"	Annotated boxes with format

EPIC_test_s1_timestamps.csv

CSV file containing 4 columns:

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 action.
stop_timestamp	string	00:23:47.212	End time in HH:mm:ss.SSS of the action.

EPIC_test_s2_timestamps.csv

CSV file containing 4 columns:

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 action.
stop_timestamp	string	00:23:47.212	End time in HH:mm:ss.SSS of the action.

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 (include

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

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.

Please go here if you wish to download the videos/frames.

License

Changelog

- 09/04/18: Initial Release