

# Recognizing Manipulation Actions from State-Transformations (Technical Report)

Nachwa Aboubakr, James L. Crowley, Rémi Ronfard

▶ **To cite this version:**

Nachwa Aboubakr, James L. Crowley, Rémi Ronfard. Recognizing Manipulation Actions from State-Transformations (Technical Report). [Research Report] Univ. Grenoble Alps, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France. 2019. hal-02197595

**HAL Id: hal-02197595**
**https://hal.science/hal-02197595**

Submitted on 30 Jul 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Recognizing Manipulation Actions from State-Transformations

Nachwa Aboubakr
Univ. Grenoble Alpes, CNRS,
Inria, Grenoble INP, LIG,
38000 Grenoble, France
nachwa.aboubakr@inria.fr

James L. Crowley
Univ. Grenoble Alpes, CNRS,
Inria, Grenoble INP, LIG,
38000 Grenoble, France
james.crowley@inria.fr

Remi Ronfard
Univ. Grenoble Alpes, Inria,
Grenoble INP, CNRS, LJK,
38000 Grenoble, France
remi.ronfard@inria.fr

## Abstract

*Manipulation actions transform objects from an initial state into a final state. In this paper, we report on the use of object state transitions as a mean for recognizing manipulation actions. Our method is inspired by the intuition that object states are visually more apparent than actions thus provide information that is complementary to spatio-temporal action recognition. We start by defining a state transition matrix that maps action verbs into a pre-state and a post-state. We extract keyframes at regular intervals from the video sequence and use these to recognize objects and object states. Change in object state are then used to predict action verbs. We report results on the EPIC kitchen action recognition challenge.*

## 1. Introduction

Most current approaches to action recognition interpret a frame sequence as a spatio-temporal signal. However, extending a 2D convolutional network by adding a 3rd temporal dimension to the receptive field results in a substantial increase in the number of parameters that must be learned, greatly increasing the computational cost and the requirements for training data. An alternative approach is to decompose recognition into a static recognition phase using a 2D kernel followed by wither a 1D temporal kernel [19], or a Recurrent Neural network [7]. Researchers have also explored the use of two-stream networks in which one stream is used to analyze image appearance from RGB images and the other represents motion from optical flow maps [18, 14, 11]. such approaches provide spatio-temporal analysis while avoiding the very large increase in learned parameters.

An alternative to learning spatio-temporal models for action recognition from video is to recognize changes in properties of objects from a sequence of frames [13, 3]. Baradel et al. [3] proposed a convolutional model that is trained to predict both object classes and action classes in two branches. This model is followed by an object relation network that learns to reason over object interactions.

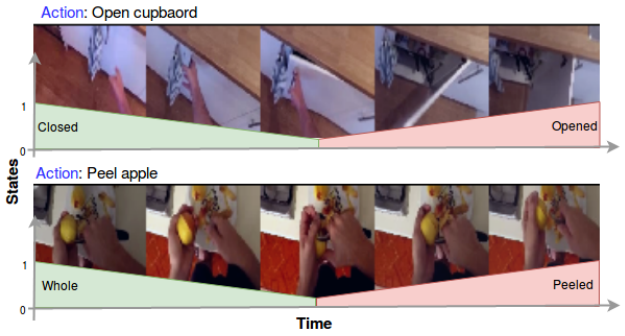



Figure 1. Changes in object states over time for action recognition. Two sample sequences from the EPIC kitchen dataset.

Our approach is inspired by the human ability to recognize changes in situation using a limited number of static observations. Human associate observations with background knowledge in a form of previously seen episodes or past experience [9, 4]. Thus a change in an object's state allows a human to form hypotheses about how the object was changed. This ability allows a human subject to interpret a complex scene from static images and make hypotheses about unseen actions that may have occurred and could explain changes to the scene. For example, we can understand which action is shown in Figure 1 with 5 keyframes or less from the video clip. Inferring the associated actions in frame sequences is a relatively effortless task for a human, while it remains challenging for machines [16]. We have investigated whether such an approach can be used to infer unseen actions from a set of frames which are chronologically ordered and contains semantic relations between objects. Such inference would complement hypotheses from spatio-temporal action recognition.

A manipulation action transforms an object from a pre-existing state (pre-state) into a new state (post-state). Thus we can say that the action *causes* a change in the state of the corresponding object. Alayrac et al. [2] have investigated

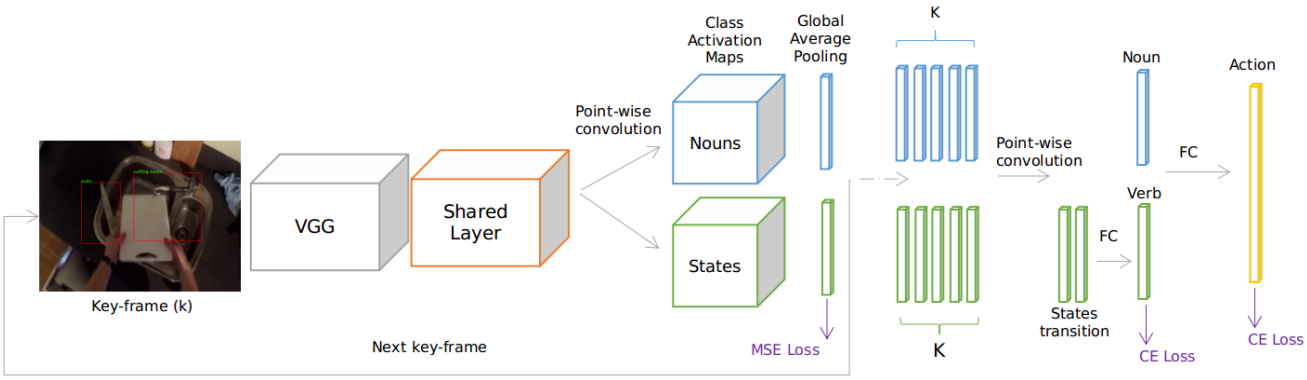



Figure 2. Proposed architecture of learning action recognition as state transformations.

automatic discovery of both object states and actions from videos. They treat this problem as a discriminative clustering problem by exploiting the ordering of the frames. Their work is promising, even though it has been evaluated on only a small number of action classes.A related work [8], studies visual changes of objects state between first and last frames.

In this paper, we investigate the feasibility of recognizing object types and object states from a small number of frames and then use changes in object states to predict actions. Our intuition is that 2D object types and states are are easier to recogninze than spatio-temporal action verb.

## 2. Manipulation action as state transformation

An action, as defined in the Cambridge dictionary[1], is the effect something has on another thing. Many manipulation actions can be expressed as triple in which a subject imparts a change to an object. That is, a manipulation action $a_i \in A$ can be expressed as: the subject that performs the action, the verb $v_i \in V$ which describes the effect of the action, and the object $n_i \in N$ the effect is applied to. For egocentric data such as EPIC kitchen the subject assumed to be the person.

The action recognition problem can be formulated with one class for each possible combination of these elements. For example, *person cuts tomato* and *person cuts cucumber* can be considered as two different classes as in [17]. Some recent datasets have provided a decomposition of an action into a verb and one or more objects $a = (v, (n_1, .., n_n))$ [10, 5, 12]. This makes it possible to study the task of action recognition as a composition of several sub-tasks (e.g. object detection and action verb recognition).

[1]Cambridge University Press. (2019). Cambridge online dictionary, Cambridge Dictionary online. Retrieved at April 3, 2019

### 2.1. State-changing actions

We are concerned with recognizing manipulation actions that change the state of objects $s_i \in S$. The state change can appear in the object's shape, its appearance (color), or its location. Examples of object states include: closed, opened, full, empty, whole, and cut.

We define a state transition function $F$ that transforms the corresponding object from a pre-state $s_i$ into a post-state $s_j$. In some cases, this state transition can be defined directly from the type of action verb $v_i$. We observe that sometimes a single verb is not enough to distinguish an action. For example, the verb *remove* can mean open in *remove lid* and can mean peel in *remove the skin of the garlic*. Therefore, the state transition must take into account both action verbs and nouns.

Since the state changes happen as we move through time, the transition function $F$ returns a real value of each state depending on the frame position in the video segment. As in Figure 1 the object starts in its initial state that gradually fades out and the post-state starts to appear as we advance in the video. In our initial experiments we have assumed that the state changing frame is the mid-frame of the video sequence. Therefore, we define the action transition mapping function $F(v, n)$, which takes the action's verb $v$ and a set of objects (nouns) $n$ and returns a continuous value of objects' states for each frame depending on the frame position in the video. For example, the action *open fridge* changes the fridge state from opened to closed.

### 2.2. Architecture

In previous work [1], we investigated detection and location of object types as well as object states from images. In this paper, we extend this work to learn changes in object state from keyframes. The architecture of our model is shown in Figure 2. Given a video segment, we

| | Seen kitchens subset (S1) | | | | Unseen kitchens subset (S2) | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc T1 | Acc T5 | Precision | Recall | Acc T1 | Acc T5 | Precision | Recall |
| | | | | Action | | | | |
| Our model(RGB) | 19.76 | 36.98 | 9.83 | **10.23** | 9.08 | 19.46 | 3.68 | 4.77 |
| 2SCNN[14](RGB) | 13.67 | 33.25 | 6.66 | 5.47 | 6.79 | 20.42 | 3.39 | 3.01 |
| TSN[18](RGB) | **19.86** | **41.89** | **9.96** | 8.81 | **10.11** | **25.33** | **4.77** | **5.67** |
| | | | | Verb | | | | |
| Our model(RGB) | **47.41** | 81.33 | 31.20 | 20.43 | 34.35 | 69.24 | 15.09 | 11.00 |
| 2SCNN[14](RGB) | 40.44 | 83.04 | 33.74 | 15.9 | 33.12 | 73.23 | 16.06 | 9.44 |
| TSN[18](RGB) | 45.68 | **85.56** | **61.64** | **23.81** | **34.89** | **74.56** | **19.48** | **11.22** |
| | | | | Noun | | | | |
| Our model(RGB) | 28.31 | 53.77 | 21.21 | 22.48 | 17.48 | 37.56 | 10.71 | 12.55 |
| 2SCNN[14](RGB) | 30.46 | 57.05 | 28.23 | 23.23 | 17.58 | 40.46 | 11.97 | 12.53 |
| TSN[18](RGB) | **36.8** | **64.19** | **34.32** | **31.62** | **21.82** | **45.34** | **14.67** | **17.24** |

Table 1. Results on the EPIC kitchen dataset (Seen and Unseen subsets). Highest values are in bold. Results of baseline methods (2SCNN and TSN) are reported by [5].

first split it into $k$ sub-segments of equal length and sample a random keyframe from each sub-segment. For each keyframe, we learn two concept classes (object types and object states) separately. Then, from the selected sequence of $k$ keyframes, we extract two channels using a point-wise convolution from which we construct the state transition matrix (pre-state, post-state). For object types (nouns), we use a point-wise convolution to extract a vector of nouns that appear in the video segment. Action verbs are then learned from the state transition matrix. In the end, the action classes are learned directly from the set of object types and action verbs.

## 3. Experiment

**EPIC Kitchen dataset.** We jave investigated state transformations using action labels using the egocentric videos of people cooking and cleaning in the EPIC Kitchen dataset. In this dataset, an action label is composed of a tuple of $a_i =$ (verb $v_i$, noun $n_i$) extracted from a narrated text given for each video action segment.

The EPIC verb represents the action verb while the EPIC noun is the action object. As the EPIC Kitchen dataset is an egocentric dataset which suggests one subject in the scene, the action subject is always the cook's hands. We group each action verb depending on the type of effect they cause into 3 different groups: those that change the object's shape, color appearance, or location. This study leaves some non-state-changing verbs (like the verb *check*) out of those groups as it does not change any object states. As a result we define 49 state transitions and 31 different states.

**Network Architecture.** As shown in Figure 2, we use a similar setting as in [1] for each keyframe. We start by extracting features using a VGG16 network with batch normalization [15] pre-trained on the ImageNet dataset [6]. VGG features provide the input to a shared[2] $3 \times 3$ convolutional layer. We separate the learning of object attributes into two branches: one for object types and the other for object states. Each attribute is learned with an independent loss. VGG features are frozen during the training process for object types and states.

For each keyframe, one noun vector and one state vector are extracted using Global Average Pooling over corresponding Class Activation Maps. Afterwards, we perform a point-wise convolution to extract one noun vector and the states transition matrix over keyframes. Verbs are learned directly from the state transition matrix using a fully-connected (FC) layer. Both action attributes (verb, nouns) are fused using at a late stage a FC layer for action classification. All hidden layers use the ReLU (rectified linear unit) activation function. A frame can have one or more states and/or nouns. Therefore, we treat nouns and states as multi-label classification problems that are learned with a Mean Square Error (MSE). On the other hand, verbs and actions are learned with a Cross Entropy (CE) function.

**Training.** We use EPIC Kitchen video segments for training our model. A clip is a collection of $k$ randomly sampled keyframes from $k$ equal length sub-segments, and it represents the corresponding action video segment. This strategy has been used in multiple works with similar problems [18, 3]. We divide the EPIC videos in 80% for training and

[2] shared over both attributes (object types and states)

| | take | put | open | close | wash | cut | mix | pour | peel | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Precision (%) | 56.7 | 59.3 | 58.8 | 39.8 | 80.1 | 74.7 | 68.9 | 39.1 | 37.7 | 57.23 |
| Recall (%) | 48.2 | 45.0 | 62.9 | 57.1 | 67.7 | 60.7 | 50.2 | 40.3 | 53.5 | 53.96 |

Table 2. Model performance on validation set on state-changing verbs.

20% for validation. Our validation set has only samples from many-shot actions and all samples of few-shot actions are in the our training split.

**EPIC challenge evaluation.** For evaluation, we aggregate the results of 10 clips as in [3]. We report the same evaluation metrics provided by the EPIC challenge [5]. Provided metrics include class-agnostic and class-aware metrics; Top-1 and Top-5 micro-accuracy in addition to precision and recall over only many shot classes (i.e. classes with more that 100 samples).

**Implementation details.** For learning, we used MSE loss to learn nouns and states during per-frame learning. Object nouns in the Actions of EPIC dataset are used to define our object classes. Each action of EPIC dataset is a tuple of a verb and a noun. The noun is chosen to be the first noun occurring in the narration sentence. Because sentences and frames can contain multiple objects, we train to detect all nouns in the sentence and treat this training step as a multi-label recognition problem for each frame. Because object state changes gradually, the state is represented as a continuous number estimated using MSE.

In training, we used the Adam optimizer with a learning rate of $1e{-}3$ that decreases following the Reduce on Plateau scheduling method. The implementation code is available[3] and was written using Pytorch.

## 4. Discussion

**Comparison with baselines.** We report the results of our model in Table 1 on EPIC Kitchen dataset for action recognition task. As the test sets are not publicly available yet, we compared our results to two baseline techniques, 2SCNN model [14] and TSN model [18], as reported in [5].

In our model, we only use RGB channels. Our model has 20M parameters and only 5M trainable parameters which is significantly lower than both baseline techniques i.e. for each input modality: 2SCNN model [14] uses 170M trainable parameters and TSN model [18] has 11M trainable parameters. Our model outperforms 2SCNN model [14] in most of reported metrics and provides recognition of verbs and actions that is comparable to TSN reported results[18].

[3]Code is available at https://github.com/Nachwa/object_states

**State-changing Actions.** In order to evaluate our model on state-changing actions, we report results of our validation set in Table 2. The model is trained to learn state changes and shows better performance on state-changing verbs than on verbs that are not state changes.

Our results show some confusion between semantically similar verbs like (e.g. insert and put, or put and move to) and verbs that have visually similar states (e.g. wash and fill - where fill examples refers to filling water from a tap). Our model is not designed to detect actions that do not result in a change in object state (e.g. move and walk).

## 5. Conclusion

In this paper, we investigated a method for recognition of manipulation actions as changes of state of objects in keyframes. We demonstrate that this can provide reasonably accurate recognition of manipulation actions. We reported results of our model on the challenge of EPIC kitchen dataset and compare these to two baseline techniques. For the action recognition task, our model outperforms one of the baseline techniques using 34 times less training parameters, and achieved comparable results with the other.

## References


[1] Nachwa Aboubakr, Remi Ronfard, and James Crowley. Recognition and localization of food in cooking videos. In *ACM International Conference Proceeding Series*, 2018. 2, 3

[2] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2127–2136, 2017. 1

[3] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 1, 3, 4

[4] Samy Blusseau, A Carboni, A Maiche, Jean-Michel Morel, and R Grompone von Gioi. A psychophysical evaluation of the a contrario detection theory. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1091–1095. IEEE, 2014. 1

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 4

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 1

[8] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586, 2013. 2

[9] François Fleuret, Ting Li, Charles Dubout, Emma K Wampler, Steven Yantis, and Donald Geman. Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences*, 108(43):17621–17625, 2011. 1

[10] Xiaofeng Gao, Ran Gong, Tianmin Shu, Xu Xie, Shu Wang, and Song-Chun Zhu. Vrkitchen: an interactive 3d virtual environment for task-oriented learning. *arXiv preprint arXiv:1903.05757*, 2019. 2

[11] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Joint learning of object and action detectors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4163–4172, 2017. 1

[12] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 619–635, 2018. 2

[13] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2018. 1

[14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1, 3, 4

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICRL)*, 2015. 3

[16] Sebastian Stabinger, Antonio Rodríguez-Sánchez, and Justus Piater. 25 years of cnns: Can we compare to human abstraction capabilities? In *International Conference on Artificial Neural Networks*, pages 380–387. Springer, 2016. 1

[17] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. ACM, 2013. 2

[18] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 1, 3, 4

[19] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatio-temporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 1