Learning Object States from Videos

Liang-Kang Huang
liangkah@andrew.cmu.edu

Katerina Fragkiadaki
katef@cs.cmu.edu

Abstract

In this work, we borrow the idea of states in automata to define states for objects that are common in our daily activities. When human interact with objects, the objects are likely to go through a series of state changes, altering their appearance and functionality from time to time. The goal of this project is to take an initial step toward enabling robots to obtain knowledge of unseen objects either by watching users‘ demonstrations or by playing around with the object’ by itself. We collected a small dataset that covers a wide range of objects with diverse property and functionality. Our result pointed out that detecting state changes is a non-trivial task and requires higher level of understanding and reasoning than simply detecting change of frame appearance.

1 Introduction

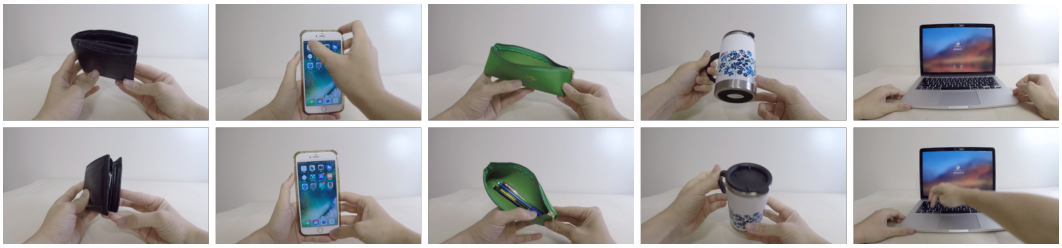
The property and functionality of objects is a fundamental knowledge for machines to understand and interact with the environment. While most of the existing works focus on object affordance (the type of interaction we could perform to an object), in this work, we explore a different aspect of object property and functionality, namely the object states.

In automata, the state of a program is determined by the content of the program's memory, or more specifically, the value of the program's internal variables. Under different states, the program reacts differently to the same set of inputs, and ends up transiting to different next-states. The program's behavior can thus be summarized by a state diagram, depicting all possible states of a program and the transition between them upon receiving different inputs. In this work, we borrow this idea from automata to define states for objects. Object states are jointly defined by the appearance and the functionality it could provide. Similar to executing a program, when we utilize an object for a specific task, the object may go through a sequence of state changes, altering its appearance and functionality from time to time. The input for triggering these changes are thus the interaction we provide to the object. See figure 1 for example

In this project, we explore the task of learning object states through video demonstrations. The major challenge is that detecting state changes is not equivalent to simply detecting scene changes or appearance change in frames. First of all, appearance change of the frames could be caused by various reasons other than change of object states. For example, users or users' body parts moving around during demonstration, users rotating/translating objects in 3D space without changing the object state, but only changing the 3D object pose relative to the camera, etc. See figure 1b for example. On the other hand, some object state changes only accompanied with subtle appearance changes that are hard to be detected by naive methods. See figure 1a for examples. Taking all these possibilities into account, we can conclude that detecting object state changes is not a trivial task. It is not clear how we could differentiate between above scenarios just by simple approaches such as applying thresholds on frame difference.

Due to lacking existing dataset suitable for our task, we collect our own dataset by recording 12 video clips demonstrating user interactions with 12 different objects. Although the dataset is small in its

11777 Course Project (2017).



(a) Examples of objects in the same state. Although the objects are in the same state, the appearance of the two frames could be very different due to change of object poses (wallet, pen-bag and mug) and movement of the hand (cell-phone and laptop).

(b) Examples of objects in the different states. Although the objects are in different states, the appearance of the two frames could be very similar due to the object is in the very beginning of a state transition (pot and ring-toy) or the change in object appearance are subtle (lamp, wallet and flap-toy)

Figure 1: Examples for illustrating the difference between detecting state changes and detecting appearance changes of frames.

size, it covers a wide range of object state changes such as turning on and off the light, unzipping the bag, putting vegetables into the pot, launching an application on the cell-phone, etc. For detail specifications of the dataset see section 3. Considering the task requires high level understanding over the scene and objects, we choose to learn the task with deep neural network models. Over the past few years, convolutional neural networks (CNN) pretrained on large scale image recognition datasets has demonstrated the ability of extracting sophisticated object-level features. We also use such a pretrained CNN for part of our model. Besides, taking into account that the information of previous and later frames could help deciding the object state in the current frame, we use a bidirectional recurrent neural network to perform sequence to sequence (frame sequence to object state sequence) prediction. For details of our approach, see section 4. We demonstrated that even with such a small and diverse dataset, it is still possible for the deep models to learn certain concepts of object state changes that could be generalized to unseen objects.

The final objective of this line of research is to allow robots to obtain knowledge of unseen objects either by watching users using the objects as in their daily activities or by playing around with the objects by the robot itself. This project takes an initial step toward this final goal by collecting a small video dataset and learn deep neural network models on top of it to see how good can these sophisticated model perform on this non-trivial task, given only limited training data.

2 Related Work

To the best of our knowledge, this is the first work explicitly defining object states and focusing on learning object states from videos. We consider two lines of research that explore other aspects of object property and functionality.

The first line is about learning and exploring object affordance. Most of the definitions for object affordance follow the one given by Gibson in [2] (first published in 1979): "properties of an object [...] that determine what actions a human can perform on them." Following this definition, many works formalize the problem of learning object functionality as identifying the possible human-object interactions for different classes of objects. Gupta et al.[4] uses a Bayesian approach to jointly

perform human pose recognition, object detection, and object functionality recognition. Grabner et al.[3] and Jian et al. [5] both take the approach of inferring object affordance by hallucinating possible human configurations in 3D spaces. Yao et al.[14] learns object functionality by a weakly supervised approach. Specifically, using existing human pose estimators and object detectors, they are able to obtain information of object affordance by analyzing the majority of human poses during human-object interactions in a bunch of training images. The above literature only learn to predict affordance for objects that belongs to classes in the training set.[15] differentiates with them by exploring how to learn a more general representation such that it could be use to predict affordance for novel object classes. Although object affordance is related to object state in the sense that we also consider the human-object interactions in videos to provide useful important information for identifying object state, the difference is that in most of the work studying object affordance, there isn't the notion that the same object could alter its appearance and functionality after certain interactions.

The other line of research is about learning object attributes. Research in this field are often closely related to object classification and recognition [9][13]. The difference is that the additional supervision of object attributes could help regularize the learning process and also related object of different classes but share common properties, so that the result is possible to generalize to unseen object classes. There are already several attribute datasets for specific categories such as scenes[10], animals[7], faces[8], etc. and also datasets that cover general objects such as [1] and [11]. Although some of the object attributes considered by these dataset are somehow related to the functionality of objects, most of the attributes are more related to the appearance, texture and other visual characteristics. However, these provide rather indirect or even no information about how the object would react during human-object interaction, which is the major focus of the object property and functionality considered in our work.

3 Dataset

Figure 2: The 12 objects in our dataset. First row: pot, tape measure, lamp, toy(rings and peg), cell phone, eye-glasses box. Second row: shaver, pencil bag, toy, wallet, mug, laptop.

	#states		#states		#states
DeskLamp	4	Mug	3	RingToy	10
FlapToy	12	PenBag	6	TapeMeasure	7
GlassesBox	6	Phone	6	Shaver	5
Laptop	6	Pot	4	Wallet	7

Table 1: List of objects and number of distinct states in their videos

We collect our own dataset by recording 12 video clips from a first person view point. See figure 2 for the list of objects. In each video, the user demonstrates several different states of an object. Ideally, we want our model be able to learn object transitions through demonstrations that are close to our daily activities. However, in this project, since we have limited number of data, we explore with a more constrained setting where users are playing with the object and demonstrating different states on purpose.

The total list of objects in the dataset could be found in figure **??**. Although the dataset is small in its size, it covers a wide range of object state changes such as turning on and off the lamp, closing a lid of the mug, unzipping the bag, putting vegetables into the pot, launching an application on the

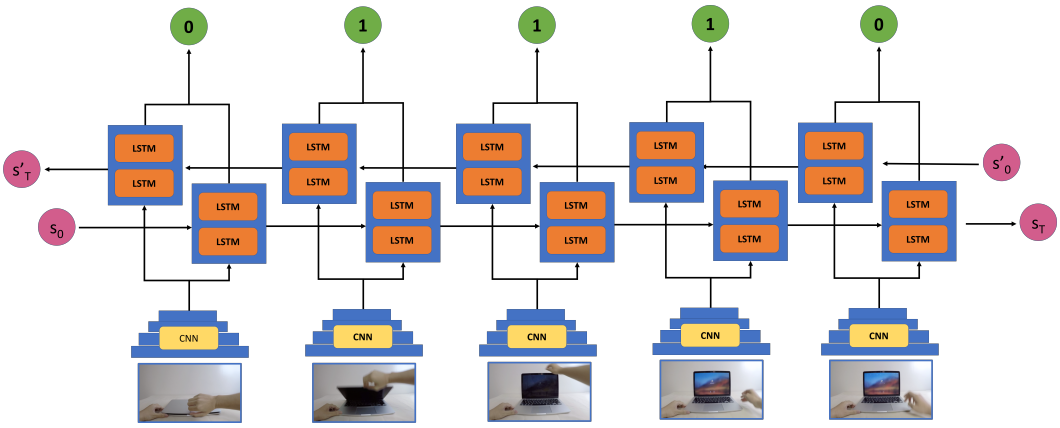


Figure 3: Illustration of the proposed model. Note that the bidirectional RNN layers and the CNN layers are sharing weights across different time steps.

cell-phone, etc. The videos are typically one to two minutes long, and demonstrate two to twelve different object states per video. For every frame in the videos, a single integer (1 to K) is assigned to it as the label of the object state number.

Annotating the state number of each frame is not a trivial task. According to the original definition, object state is jointly defined by object appearance and functionality. Which means we should consider two states as different whenever the objects do not have the exact same appearance or functionality. However, strictly following this guideline may result in too many state transitions in the video and accounting for too many subtle states. Thus, we have special treatments for the following two cases. The first case is when the object is continually changing its state or functionality in a short period of time. For example, when the bag is being unzipped, or a laptop being opened. For these cases, instead of assigning a new state for every next frame, we simply annotate these frame as -1, indicating that they are in the middle of state transitions. The second case is when objects only have minor changes in appearance or functionality. For example, launching a calculator application on the smart phone is considered a state change, but user typing a digit on the calculator is not. Another example is an pencil bag with two pens inside is considered to be in a different state compared with the same bag with two pens and a ruler. However, when the two pens are placed in slightly different position in the bag, or say if they exchange their positions with each other should not cause state changes to the bag.

4 Method

In this project, we formulate our task into a supervised learning problem. The dataset described in section 3 contains labels of object state for every frame in the videos. Given this information, we can come up with a new label for each frame, indicating whether the object state remains the same from the previous frame to this frame. In other words, instead of directly learning to predict the current object state, our model learns to decide whether there is a state change from the previous frame to the current frame. In addition, we believe that in order to perform this inference, the essential information resides in a short interval of the target frame. Simply considering the information in a pair of frames may not be sufficient to determine whether there exists a state change. For this reason, each of our training instance consists of five consecutive frames. The desired output is then a binary sequence of length four, indicating whether the there is a change of object state in the last four frames. (We do not do the prediction for the first frame given that there is no information for the previous frame in this case). The final problem thus becomes a sequence to sequence prediction where the input is a sequence of frames and the output is a sequence of binary numbers.

For the model architecture, We choose to concatenate a convolution neural network(CNN) with a recurrent neural network(RNN) as our final model. The reason for using CNN is due to its proven ability for capturing sophisticated features of objects and object parts. We believe that detecting state

	Precision	Recall	F1
Frame Difference	0.312	0.561	0.401
Ours	0.865	0.711	0.781

(a) PenBag

	Precision	Recall	F1
Frame Difference	0.253	0.388	0.306
Ours	0.490	0.848	0.604

(b) Wallet

Table 2: Quantitative results comparing the baseline model and our approach on the PenBag and Wallet sequence.

changes, unlike detecting appearance change, requires high level object features to be extracted. We use the popular AlexNet[6] architecture pretrained on the ILSVRC 2015 dataset[12], and remove the last three fully-connected layers (so the extracted feature vectors contain the spatial information of the objects). Given N consecutive frames F_1, F_2, ... , F_N, we pass them thourgh the same CNN to generate the feature vectors x_1, x_2, ... ,x_n.

$$x_t = f_C(F_t), t = 1, 2, ..., N$$

The RNN is used for performing the sequence to sequence prediction. We choose to use bidirectional RNN since we believe better decision could be made for the target frame if information in both the previous and later frames are considered at the same time. Also the RNN is equipped with a two layer LSTM cell in order to remember selective contents of observed frames. See figure 3 for an overview of our model. Given the feature vectors x_1, x_2, ... ,x_n, the bidirectional RNN predicts state changes y_1, y_2, ... ,y_n with the following formulation:

$$\begin{aligned}
\vec{h}_t &= f_R(x_t, \vec{h}_{t-1}) \\
\overleftarrow{h}_t &= f_R(x_t, \overleftarrow{h}_{t-1}) \\
y_t &= g(W[\vec{h}_t; \overleftarrow{h}_t] + b)
\end{aligned}$$

where $\vec{h}_t$ and $\overleftarrow{h}_t$ are the hidden states of the bidirectional RNN at time step t. The final output y_t is generated by a fully connected layer with the input being the concatenation of $\vec{h}_t$ and $\overleftarrow{h}_t$.

After prediction, we need a post processing step to aggregate the prediction result of every five frames into the final prediction for the full video. The idea of this post processing is to provide temporal smoothing for the final prediction, which is not guaranteed if the predictions for every five consecutive frames are performed independently. In our case, we use a weighted sum of the prediction result, with the weight determined by a Gaussian kernel centered on the target frame. After the weighted sum, a median filter of window size three is performed to eliminate outliers and noisy predictions to generate the final result.

5 Result

In our experiment, we use 10 videos for training, and the rest 2 videos (PenBag and Wallet) for testing. Here we compare to the baseline approach of detecting state changes by thresholding frame difference. The threshold is picked by searching a range of values and use the one that gives the best performance on the training set. Quantitative results are reported in table 2 and although the qualitative results are best view in video forms, in figure 4 and 5 we sample a short sequence of frames from the videos to present the result.

PenBag In this sequence, our model performs pretty well on most part of the sequence. Although there is no non-rigid object like this the model doesn't confuse the non-rigid transformations with

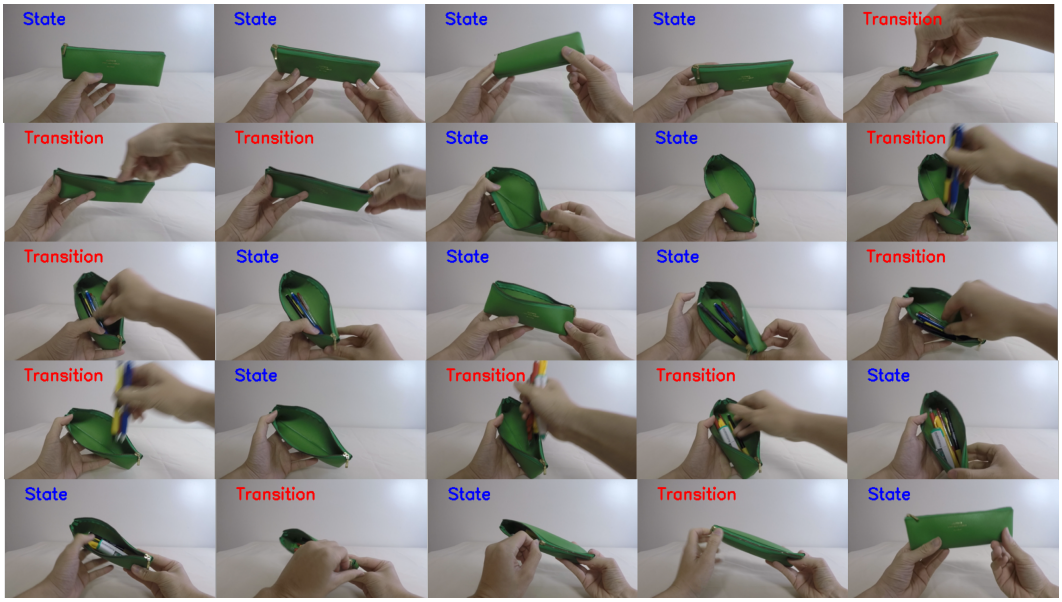


Figure 4: Illustration of the proposed model. Note that the bidirectional RNN layers and the CNN layers are sharing weights across different time steps.

Figure 5: Illustration of the proposed model. Note that the bidirectional RNN layers and the CNN layers are sharing weights across different time steps.

object state changes when the bag is squeezed during demonstration. Also in the training set we don't have similar action such as zipping and unzipping the zipper, however the model is able to generalize the learning result to these actions and recognized that the bag is going through state transitions during these actions.

Wallet In this sequence, our model performs slightly worse than the PenBag sequence, mis-predicting two important state changes: First is when the wallet is opened (see the last two frames in the first row of figure 5) the model doesn't recognize the state change here. The second is when wallet is opened to show the bills inside (see the second to last frame in the last row of figure 5). In the sequence, the user stayed in this state for several seconds but the model still predict all the frames in this interval to be in the middle of a transition. For the rest of the transitions like flipping the inner cover (see the last three frames of the second row of figure 5), taking out the metro card (see the first three frames in the third row of figure 5) and opening the inner pocket (see the fourth row of figure 5), the model can distinguish them from the rigid transformation of wallet performed by the users from time to time.

6 Conclusion

This work takes a first step toward learning object states through video demonstrations. The contribution of this work includes presenting the idea of object states, identifying the challenges of recognizing object states, collecting the first object state dataset and also trying out state of the art methods to explore their effectiveness on this task. Futre work includes augmenting the dataset to contain more objects and also moving toward videos that are more close to how human interact with objects in daily activities. We expect our work to draw interest in a different aspect of object property and functionality, together with the workds studying object affordance and atttributes, allow machines to have a much richer understanding over objects and the environment.

References

[1] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR*, 2009.

[2] J.J. Gibson. *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press & Routledge Classic Editions. Taylor & Francis, 2014.

[3] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *CVPR*, 2011.

[4] Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1775–1789, 2009.

[5] Yun Jiang, Hema Koppula, and Ashutosh Saxena. Hallucinated humans as the hidden context for labeling 3d scenes. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2993–3000, Washington, DC, USA, 2013. IEEE Computer Society.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by betweenclass attribute transfer. In *In CVPR*, 2009.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[9] Wentao Luan, Yezhou Yang, Cornelia Fermüller, and John S. Baras. Reliable attribute-based object recognition using high predictive value classifiers. *CoRR*, abs/1609.03619, 2016.

[10] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proceeding of the 25th Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] Genevieve Patterson and James Hays. Coco attributes: Attributes for people, animals, and objects. *European Conference on Computer Vision*, 2016.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[13] Xiaoyang Wang and Qiang Ji. Object recognition with hidden attributes. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3498–3504. AAAI Press, 2016.

[14] Bangpeng Yao, Jiayuan Ma, and Li Fei-fei. Discovering object functionality.

[15] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV (2)*, volume 8690 of *Lecture Notes in Computer Science*, pages 408–424. Springer, 2014.