The “something something” video database for learning and evaluating visual common sense

Raghav Goyal
raghav.goyal@twentybn.com

Samira Ebrahimi Kahou
samira.ebrahimi.kahou@gmail.com

Vincent Michalski
michalskivince@gmail.com

Joanna Materzyńska
joanna.materzynska@twentybn.com

Susanne Westphal
susanne.westphal@twentybn.com

Heuna Kim
heuna.kim@twentybn.com

Valentin Haenel
valentin.haenel@twentybn.com

Ingo Fruend
ingo.fruend@twentybn.com

Peter Yianilos
peter@yianilos.com

Moritz Mueller-Freitag
moritz.mueller-freitag@twentybn.com

Florian Hoppe
florian.hoppe@twentybn.com

Christian Thurau
christian.thurau@twentybn.com

Ingo Bax
ingo.bax@twentybn.com

Roland Memisevic
roland.memisevic@twentybn.com

Abstract

Neural networks trained on datasets such as ImageNet have led to major advances in visual object classification. One obstacle that prevents networks from reasoning more deeply about complex scenes and situations, and from integrating visual knowledge with natural language, like humans do, is their lack of common sense knowledge about the physical world. Videos, unlike still images, contain a wealth of detailed information about the physical world. However, most labelled video datasets represent high-level concepts rather than detailed physical aspects about actions and scenes. In this work, we describe our ongoing collection of the “something-something” database of video prediction tasks whose solutions require a common sense understanding of the depicted situation. The database currently contains more than 100,000 videos across 174 classes, which are defined as caption-templates. We also describe the challenges in crowd-sourcing this data at scale.

1. Introduction

Datasets and challenges like ImageNet [3] have been major contributors to the recent dramatic improvements in neural network based object recognition [14, 30, 8], as well as to improvements on a variety of other vision tasks thanks to transfer learning (eg., [4, 27, 19]).

Despite their representational power, neural networks trained on still images ignore of a wide range of scene aspects, many of which are could be inferable from video. These include 3-D geometry (which can reveal itself through multiple views [7]), material properties (such as deformability, elasticity, stiffness, etc.), articulation, affordances [34] or intuitive physics (for example, occlusion/object permanence, gravity).

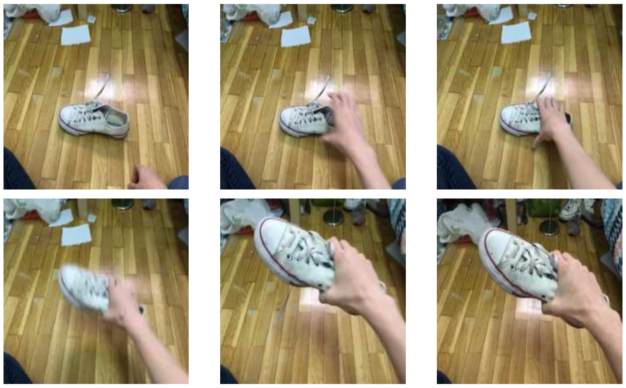


Figure 1: An example video from our database, captioned “Picking [something] up”. Crowd-workers are asked to record videos and to complete caption-templates, by providing appropriate input-text for placeholders. In this example, the text provided for placeholder “something” is *“a shoe”*. We plan to increase the complexity and sophistication of caption-templates over time, to the degree that models succeed at making predictions.

Motion patterns extracted from a video are not only capable of revealing object properties but also of revealing actions and activities. Not surprisingly, most of the currently popular labeled video datasets are action recognition datasets [26, 17, 23, 12]. It is important to note, however, that in a fine-grained understanding of visual concepts that goes beyond "one-of-K"-labeling, actions and objects are naturally intertwined, and the tasks of predicting one cannot be treated independently of predicting the other. For example, the phrase "opening NOUN" will have drastically different visual counterparts, depending on whether "NOUN" in this phrase is replaced by "door", "zipper", "blinds", "bag", or "mouth". There are also commonalities between these instances of "opening", like the fact that parts are moved to the sides giving way to what is behind. It is, of course, exactly these commonalities which define the concept of "opening". So a true understanding of the underlying meaning of the action word "opening" would require the ability to generalize across these different cases. A proper understanding of such concepts is closely related to affordances. For example, the fact that a door *can* be opened is much more likely to be taken into consideration, or even learnable, by a robot searching for an object, if its feature space is already structured such that it can distinguish between opening and closing doors.

Finally, not only words for objects and actions can be grounded in the visual world, but also many abstract concepts, because these are built by means of analogy on top of more basic, every-day concepts [15, 10]. We believe that visual grounding through video, to the degree that it can be advanced, may ultimately become a building block for language modeling and other areas in AI that appear to be non-visual at their surface.

In this work, we describe our ongoing efforts in generating the "*something something*"-database, whose purpose is to provide visual (and partially acoustic) counterparts of simple, everyday aspects of the world. The goal of this data is to encourage networks to develop the features required for making predictions which, by definition, involve certain aspects of common sense information. The growing database[1] currently contains $108,499$ *short* video clips (with duration $\in [2,6]$ seconds), that are labeled with simple textual descriptions. The videos show objects and actions performed on them. Labels are in textual form and represent detailed information about the objects and actions as well as other relevant information. Predicting labels from the videos requires features that are capable of representing physical properties of the objects and the world.

2. Related work

Although images still largely dominate research in visual deep learning, a variety of sizeable labeled video datasets have been introduced in recent years. As mentioned, the dominant application domain so far has been action recognition, where the task is to predict a global action label for a given video (for example, [23, 12, 17, 11, 5]). A drawback of action recognition is that it is targeted at fairly high-level aspects of videos and therefore does not encourage a network to learn about motion primitives that can encode object properties and intuitive physics. For example, the task associated with the datasets described in [23, 12] is recognizing sports, and in [17] they include high-level, human-centered activities, such as "getting out of a car" or "fighting". A related issue is that, in many cases, good classification performance can be achieved on these tasks using features extracted with a convolutional network (pre-)trained on still images [36].

Detailed labeling has been addressed also in various video captioning datasets recently, where the goal is to predict an elaborate description, rather than a single label, for a video [31, 24, 37, 13]. However, similar to many of the action recognition datasets mentioned above, they typically contain descriptions that reflect high-level, cultural aspects of human life and commonly require a good knowledge of rare or unusual facts and language. Recently, [38, 9] showed how captioning models can "cheat" by generating sensible sentences without necessarily understanding an observed scene in detail.

A dataset focussing on lower-level, more physical concepts is described in [35]. The dataset contains $17,408$ videos of a small set of objects involved in a number of physical experiments. These include, for example, letting the object slide down a slope or dropping it onto a surface. The supervision signal is given by (known) physical properties of the experiment, such as the angle of the slope or the material of the object. In contrast to that work, besides scaling to a much larger size, we use language as labels, similar to captioning datasets. This allows us to generate a much larger and more varied set of actions and labels. It also allows us to go beyond a small and highly specialized set of physical properties and actions prescribed by the experimental setup and by what can easily be measured.

Many shortcomings of existing video datasets may be related to the fact that they are generated by annotating (or using closed captionings of) existing video material, including excerpts from Hollywood movies. Recently, [28] proposed a way to overcome this problem by asking crowdworkers to record videos themselves rather than to attach labels to existing videos. In this work, we follow a similar approach using a scalable framework for crowd-sourced video recording. Our crowd sourcing framework allowed us to generate several hundred thousand videos so far, including the dataset discussed in this paper. In contrast to the dataset described in [28] we focus here on basic, physical concepts rather than on higher-level human activities.

[1] We made a version of the dataset available at: https://www.twentybn.com/datasets/something-something

Dataset	Domain	# Videos	Avg. duration	Remarks
Physics 101 [35]	intuitive physics	17,408	-	101 objects with 4 different scenarios (ramp, spring, fall, liquid)
MPII cooking [25]	action (cooking)	44	600s	-
TACoS [22]	action (cooking)	127	360s	-
Charades [28]	action (human)	10, 000	30s	-
KITTI [6]	action (driving)	21	30s	-
Something-Something (ours)	human-object interaction	108,499	4.03s	174 fine-grained categories of human-object interaction scenarios

Table 1: Comparison of video datasets recorded specifically for training models (information taken partially from [13])

A comparison with existing similar datasets is shown in Table 1.

2.1. Learning intuitive physics

There has been an increasing interest recently in learning representations of physical aspects of the world using neural networks. Such representations are commonly referred to as intuitive or naive physics to contrast them with the symbolic/mathematical descriptions of the world developed in physics. Several recent papers address learning intuitive physics by using physical interactions (robotics) [20, 1]. A possible shortcoming of this line of work is that it is based on using still images, which show, for example, how objects appear before and after performing a certain action. Physical predictions are made using convolutional networks applied to the images. Any sequential information is thus reduced to predicting a causal relationship between action and observations in a single feedforward computation, and any information encoded in the motion itself is lost.

There has been a long-standing endeavor to use future frames of a video as "free" labels for supervised training of neural networks. See, for example, [18, 21] and references therein. Unfortunately, predicting raw pixels is challenging, both for computational and for statistical reasons. There are simply a lot of aspects of the real world that a predictor of raw pixels has to account for. This may be one reason why unsupervised learning through video prediction has, like unsupervised learning in general, not let do the breakthrough that many have been hoping for.

A hybrid between learning from video and learning from interactions is the work by [16] who use a game engine to render block towers that collapse. A convolutional network is then trained to predict, using an image of the tower as input, whether it will collapse or not, as well as the trajectories of parts while the tower collapses. Similar to [20, 1], predictions are based on still images not videos.

3. The "something-something" dataset

In this work, we introduce the "something-something"-dataset. It currently contains $108,499$ videos across 174 labels, with duration ranging from 2 to 6 seconds. Labels are textual descriptions based on templates, such as "Dropping [*something*] into [*something*]" containing slots ("[*something*]") that serve as placeholders for objects. Crowd-workers provide videos where they act out the templates. They choose the objects to perform the actions on and enter the noun-phrase describing the objects when uploading the videos.

Dataset Specifications	
Number of videos	108,499
Number of class labels	174
Average duration of videos (in seconds)	4.03
Average number of videos per class	620

Table 2: Dataset summary

The dataset is split into train, validation and test-sets in the ratio of 8:1:1. The splits were created so as to ensure that all videos provided by the same worker occur only in one split (train, validation, or test). See Table 2 for some summary information about the dataset.

Including differences in case, stemming, use of determiners, etc., the current version of the dataset containts $23,137$ distinct object names. We estimate the number of actually distinct objects to be at least a few thousand. Figure 3 (bottom) shows the frequency of objects for the most common objects.

In its current version, the dataset was generated by 1133 crowd workers with an average of 127.32 workers per class. Figure 2 shows a truncated distribution of the number of videos per class, with an average of roughly 620 videos per class, a minimum of 77 for "Poking a hole into [*some substance*]" and a maximum of 986 for "Holding [*something*]". Figure 3 (top) shows a histogram of the duration of videos (in seconds). A few examples of frame samples from the collected videos is shown in Figure 4.

3.1. Crowdsourced video recording

The currently pre-dominant way of creating large, labeled datasets is to start by gathering a large collection of

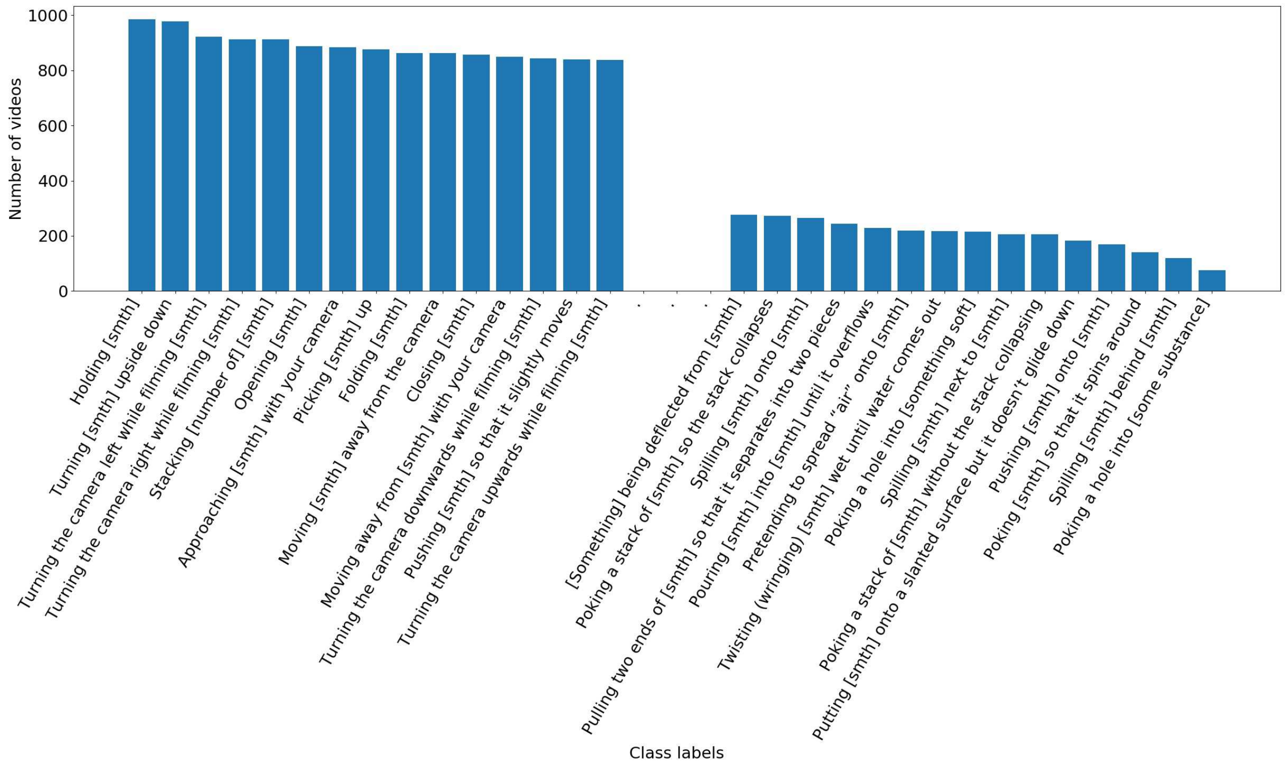


Figure 2: Numbers of videos per class (truncated for better visualisation).

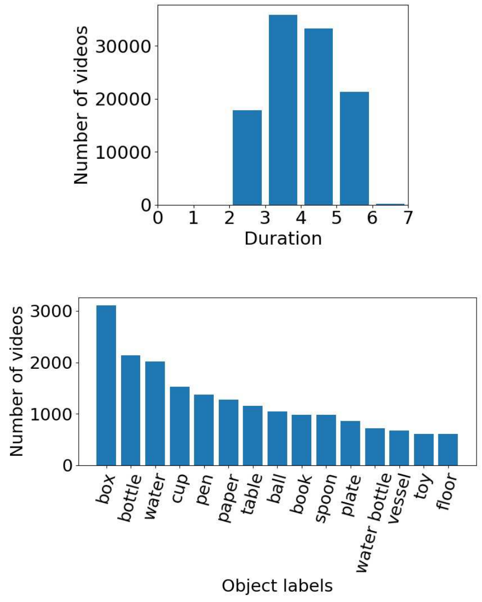


Figure 3: Top: Video lengths (in seconds). Bottom: Frequencies of occurrence of 15 most common objects

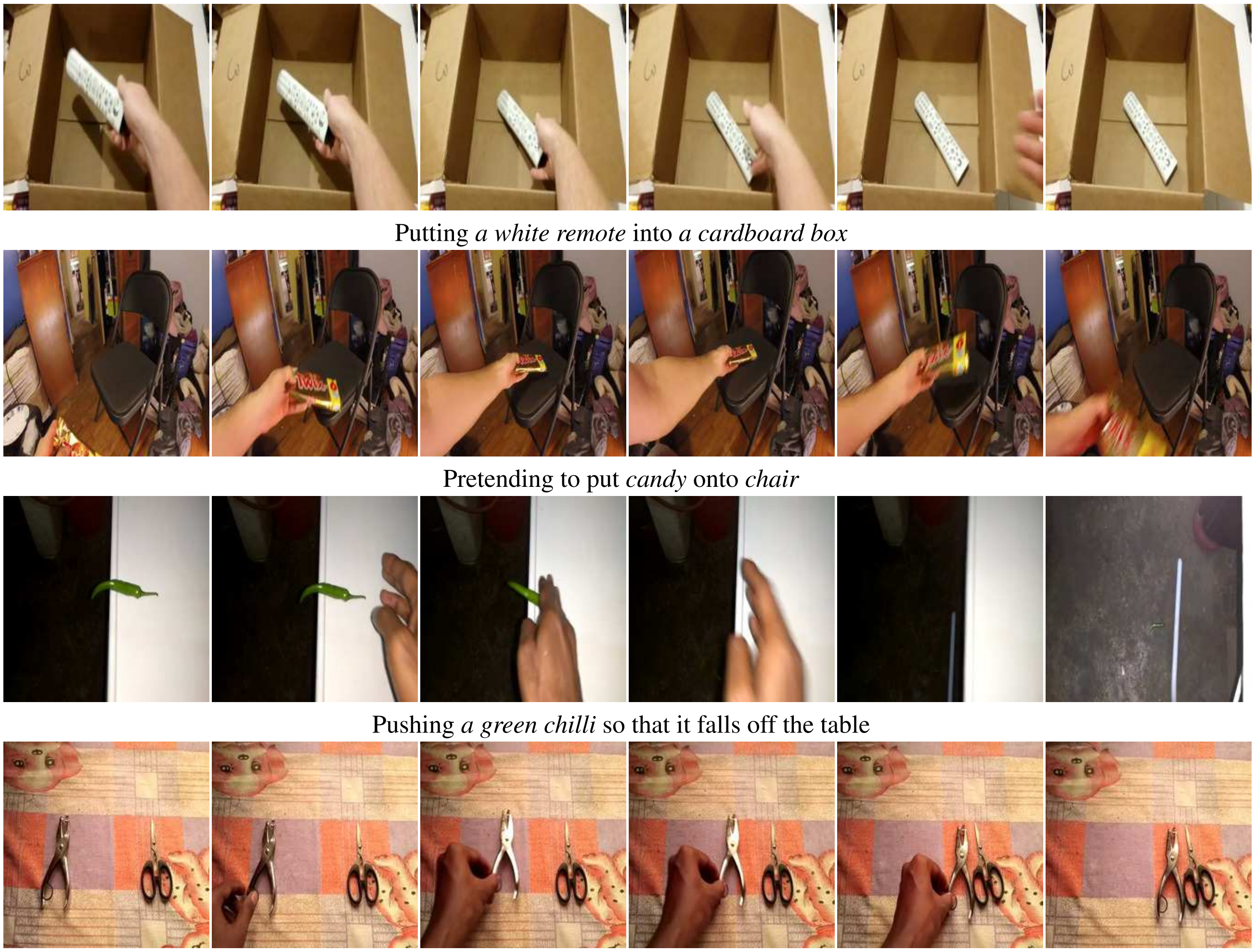
input items, such as images or videos. Usually, these are found using online resources, such as Google image search or Youtube. Subsequently, the gathered input examples are labeled using crowdsourcing services like Amazon Mechanical Turk (see, for example, [3]).

As outlined is Section 2 videos available online are largely unsuitable for the goal of learning simple (but fine-grained) visual concepts. We therefore ask crowd workers to provide videos *given* labels instead of the other way around (a similar approach was recently described in [28]).

3.2. Natural language and curriculum learning

The number of "everyday concepts" that we want to capture with this dataset is huge, and it cannot be captured within a fixed set of one-hot labels. Natural language descriptions are a natural and obvious solution to this problem: natural language is capable of representing an extremely large number of "classes" and it is compositional and thereby able to express this number highly economically.

Unfortunately, natural language provides a much weaker learning signal than a one-hot label. This is one reason why image and video captioning systems are currently trained using an ImageNet pre-trained network as the vision component.



Putting *a white remote* into *a cardboard box*

Pretending to put *candy* onto *chair*

Pushing *a green chilli* so that it falls off the table

Moving *puncher* closer to *scissor*

Figure 4: Example videos and corresponding descriptions. Object entries shown in italics.

To obtain useful natural-language labels, but also be able to train, and potentially bootstrap, networks to learn from the data, we generate natural language descriptions automatically by appropriately combining partly pre-defined, and partly human-generated, parts of speech. Natural language descriptions take the form of *templates* that crowd workers provide along with videos, as we shall describe in the next section. Analogous to how probabilistic graphical models impose independence assumptions on a multivariate distribution, these "structured captions", can be viewed as approximations to full natural language descriptions, that allow us to control the complexity of learning by imposing a rich (but restricted) structure on the labels.

In the current version of the dataset, we emphasize short and simple descriptions, most of which contain only the most important parts of speech, such as verbs, nouns and prepositions. This choice was made, because common neural networks are not yet able to represent elaborate captions and high-level concepts.

However, it is possible to increase the degree of complexity as well as the sophistication of language over time as the dataset grows. This approach can be viewed as "curriculum learning" [2], where simple concepts are taught first, and more complicated concepts are added progressively over time. From this perspective, the level of complexity of the current version of the dataset may be viewed approximately as "teaching a one-year-old child". Unlike labels that are encoded using a fixed datatype, as described, for example, in [39], natural language labels allow us to represent a spectrum of complexity, from simple objects and actions encoded as one-hot labels, to full-fledged captions. The use of natural language encodings for classes furthermore allows us to dynamically adjust the label structure in response to how learning progresses. In other words, the complexity of videos and natural language descriptions can be increased as a function of the validation-accuracy achievable by networks trained on the data so far.

3.3. Sampling action-object combinations

Although it is more restricted than captions, the Cartesian product of actions and objects constitutes a space that is so large that there is no hope to sample it sufficiently densely as needed for practical applications. But the empirical probability density of real-world cases in the space of permissible actions and objects is far from uniform. Many actions, such as "Moving an elephant on the table" or "Pouring paper from a cup", for example, have almost zero density. And more reasonable combinations can still have highly variable probabilities. Consider, for example, "drinking from a plastic bag" (highly rare) vs. "dropping a piece of paper" (highly common).

It is possible to exploit the low entropy of this distribution, by using the following sampling scheme: Each crowd worker is presented with an action in the form of a template that contains one or several placeholders for objects. Workers then get to decide which objects to perform the action on and generate a video clip. When uploading the video, workers are required to enter their object choice(s) into a provided mask.

3.4. Grouping and contrastive examples

The goal of the "something-something" collection effort is to provide fine-grained discrimination tasks, whose solution will require a fairly deep understanding of the physical world. However, especially in the early stage, where simple descriptions focussed on verbs and nouns dominate, networks can learn to "cheat", for example, by extracting the object type from one or several individual frames, and by extracting the action using indirect cues, such as hand position, overall velocity, camera shake, etc. This is an example of *dataset bias* [32].

As a way to reduce bias, by forcing networks to classify the actual actions and the underlying physics, we provide *action groups* for most action types. An action group contains multiple similar actions with minor visual differences, so that fine-grained understanding of the activity is required to distinguish the actions within a group. Providing action groups to the crowd workers also encourages these to perform the multiple different actions with the same object, such that a close attention to detail is required to correctly identify the action within the group. We found that action groups also serve the communication with crowdworkers in clarifying to them the kinds of fine-grained distinctions in the uploaded videos we expect.

Some groups contain *pretending* actions in addition to the actual action to be performed. This will require any system training on this data to closely observe the object instead of secondary cues such as hand positions. It will also require the networks to learn and represent indirect visual cues, such as the fact that an object is present or not present in a particular region in the image. Preventing a network from "cheating" by distinguishing between actual and pretended actions is reminiscent of teaching a child by asking it to tell the difference between genuine and false actions. Examples of action groups we use include:

- Putting *something* on top of *something* / Putting *something* next to *something* / Putting *something* behind *something*
- Putting *something* behind *something* / Pretending to put *something* behind *something* (but not actually leaving it there)
- Poking *something* so lightly that it does not or almost does not move / Poking *something* so it slightly moves / Poking *something* so that it falls over.
- Poking *something* / Pretending to poke *something*

A more comprehensive list of action groups and descriptions examples are provided in the supplementary materials.

3.5. Data collection platform

Besides the requirements outlined above, crowdsourcing the recording of video data according to a pre-defined label structure poses a variety of technical challenges:

- *Batch submission:* Crowd workers need to be able to initiate a job, and come back to it later potentially multiple times until it is completed, so that they can record videos outside or at other places or times of the day, or after having gathered the objects needed for the task.
- *Worker-conditional choice of labels:* To generate data with sufficient variability, it is important that each label is represented by videos from as many different crowdworkers as possible. To this end, it is necessary to keep track of the set of labels recorded by each individual crowdworker. 'The list of labels or action groups (as defined below) to choose from can be generated dynamically once the crowdworker logs on to the platform.
- *Feedback on completed or partially completed submissions:* In the case of submissions that are fully or partially rejected it is important that the crowd sourcing operators can quickly provide feedback to the crowd workers regarding what was wrong with the submission.
- *Convenience:* To reduce cost, crowd workers need to face a convenient, easy-to-use and highly responsive interface.

To address these challenges, we created a data collection platform, with which both crowd workers and our operators overseeing the crowdsourcing efforts interact during the ongoing crowdsourcing operation.

When a crowdworker accepts a task he/she gets redirected to our platform, where the task is then completed and reviewed. After completion of a task, our platform communicates the result back to the crowdsourcing service.

On the platform, workers get presented with a list of action-templates to choose from (with action-templates grouped as described in the previous section). By selecting action-templates, the platform creates video upload-boxes where workers can upload the videos as required, along with label-templates with variable-roles to be filled by workers.

After uploading a video, all variable-roles in the label template (represented by the word "something" in most of our label templates) turn into input masks, and the worker is asked to fill in the correct word (such as the noun describing the object used). Each uploaded video is displayed (as screenshot) in a video playback-box and it can be played back for easy inspection by the workers (as well as by the operators as we describe below). After the worker reaches the number of requested videos, a button "Submit Hit" gets released, that allows the worker to submit the assignment and get paid.

A submission is accepted automatically, if it passes a number of quality control checks, which verify aspects such as length and uniqueness of the videos. Every submission is subsequently verified for correctness by a human operator. For more details on the crowd acting platform and screenshots we refer to the supplementary materials.

4. Baseline experiments

We performed a few baseline experiments to assess the difficulty of the task of predicting label templates from the videos. In this work, we discuss classification tasks on the label templates. Full captioning and performance on the expanded labels will be discussed elsewhere. On the classification tasks, we found 3d-convolutional networks to generally outperform 2d-convolutional networks and their combination to work best. But we also found that many of the subtle classes that were chosen explicitly to make the task harder (Section 3.4), are hardly distinguishable using these fairly standard architectures. More sophisticated architectures are necessary to obtain better performance on this data. A difficulty for both training and interpreting results is the presence of ambiguities in the labels. For reporting, these can be dealt with to some degree by resorting to top-K error rate. Both ambiguities and the overall difficulty of the prediction tasks can be alleviated by choosing label subsets and by combining labels into groups, which can allow fairly simple architectures to achieve reasonable performance. We shall discuss several such simplified subsets of classes below. We also found that this grouping can help as an initialization for networks that are subsequently fine-tuned on more complex class-choices.

10 selected classes
Dropping [something]
Moving [something] from right to left
Moving [something] from left to right
Picking [something] up
Putting [something]
Poking [something]
Tearing [something]
Pouring [something]
Holding [something]
Showing [something] (almost no hand)

Table 3: Subset of 10 hand-chosen "easy" classes.

4.1. Pre-processing

For the baseline runs, we sample frames from the videos using a frame rate of 24 fps and resize them to a resolution of 84 × 84 pixels, except for those runs where we use a pre-trained model (in which case we use the resolution is determined by that model). We lowpass-filter the resulting videos in time using a Gaussian kernel with zero mean and variance of 48 pixels, which was chosen to largely eliminate frequencies above the Nyquist-frequency, taking into consideration the target frame-rate of 6 frames per second (as discussed below).

We also perform temporal augmentation by choosing a random offset between 0 and the downsampling factor (4) during training. We use a fixed offset of 0 for validation and testing. We have also experimented with other types of data augmentation including flipping frames for invariant classes and random rotation by a small angle, but we did not find any significant performance gains for these.

4.2. Model specifications

Here we report results on the task of predicting action templates using multiple different encoding methods. We found dropout on the first fully-connected layer and batch-normalization on the last layer to significantly improve training. The encoding methods we used are:

2D-CNN + Avg: Using the VGG-16 net architecture [29] to represent individual frames and averaging the obtained features for each frame in the video to form the final encoding. The weights of the network were trained from scratch.

Pre-2D-CNN + Avg: Using an Imagenet-trained VGG-16 architecture to represent individual frames and averaging the obtained features for each frame in the video to form the final encoding.

Pre-2D-CNN + LSTM: Using the above pre-trained VGG network to represent individual frames and passing the extracted features to an LSTM layer with a hidden state size of 256. The last hidden state of the LSTM is then taken

Method	**Error rate (%)**						
	10 classes		40 classes		174 classes		
	top-1	top-2	top-1	top-2	top-1	top-2	top-5
2D CNN + Avg	76.5	58.9	88.0	78.5	-	-	-
Pre-2D CNN + Avg	54.7	39.0	79.2	70.0	-	-	-
Pre-2D CNN + LSTM	52.3	34.1	77.8	68.0	-	-	-
3D CNN + Stack	58.1	38.7	70.3	57.3	-	-	-
Pre-3D CNN + Avg	47.5	29.2	66.2	52.7	88.5	81.5	70.0
2D+3D-CNN	**44.9**	**27.1**	**63.8**	**50.7**	-	-	-

Table 4: Error rates on different subsets of the data.

as the video encoding.

3D-CNN + Stack: Using a 3D-CNN model trained from scratch with specifications following [33], but with a size of 1024 units for the fully-connected layers and a clip size of 9 frames. We extract these features from non-overlapping clips of size 9 frames (after padding all videos to a maximal length of 36 frames), and stack the obtained features to obtain a 4096 dimensional representation (4 columns), masking the column features, such that invalid frames (due to padding) do not affect training.

Pre-3D-CNN + Avg: Using a 3D-CNN model initialized on the sports-1m dataset [33] and finetuned on our dataset. In this case, we use the framerate 8 fps for training and extract columns of size 16 frames with 8 frames overlap between columns, such that the total number of columns is 5. We average the features across the clips.

2D+3D-CNN: A combination of the best performing 2D-CNN and 3D-CNN trained models, obtained by concatenating the two resulting video-encodings.

4.3. Results

We compared these networks mainly on two subsets of the dataset with classes hand-picked to simplify the task and benchmark the complexity of the dataset (we refer to the supplementary materials for more details on selection of classes): **10 selected classes:** We first pre-select 41 "easy" classes. We then generate 10 classes to train the networks (shown in Table 3), where each class is formed by grouping together one or more of the original 41 classes with similar semantics. The mapping from 41 to 10 classes is shown in the appendix. The total number of videos in this case is 28198. **40 selected classes:** Keeping the above 10 groups, we select 30 additional common classes. The total number of samples in this case is 53267. Some example predictions from the 10-class model are shown in the appendix.

We show the error rates for these subsets using the baselines described above in Table 4. It shows that the difficulty of the task grows significantly as the number of classes are increased (despite the corresponding growth of the training-set). Similar to datasets like Imagenet, ambiguities in the labels make the naive classification performance look deceptively weak. However, even the top-2 performance shows that there the dataset poses a significant challenge for these architectures.

We also experimented on **all 174 classes** using a 3D CNN model pre-trained on the 40 selected classes, and obtained error rates of top-1: 88.5%, top-5: 70.3%.

Overall, our results demonstrate that the presence of subtle distinctions (using grouping, contrastive examples, etc.) makes this an extraordinarily difficult problem for standard architectures. We also performed an informal human evaluation on the complete dataset (174 classes) with 10 individuals who classified $\sim$ 700 test samples in total, achieving an accuracy of $\sim$ 60%. This shows that despite its difficulty and the presence of ambiguities, there is a huge potential for further research and modeling to improve the accuracy.

5. Discussion

Advances in common sense reasoning can come mainly from two sources: through learning from interactions with the world, and through learning from observing the world. The first, interactions, rely crucially on advances in robotics. Unlike human interactions, however, robotic interactions lack sophisticated tactile sensing (which allows human to learn about the world even without any vision). It therefore is likely that even a robotics-based approach to learning common sense will rely on highly capable visual perception and on visuomotor policies that can deal with video input.

This work falls into the second category: learning about the world through vision. In contrast to unsupervised approaches, based on video-prediction, we propose approaching the problem through supervised learning on fine-grained labeling tasks. The database introduced in this paper is an ongoing collection effort. We will continue to grow and extend it over time as a function of the ability of networks to learn from the data.

References

[1] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in NIPS*, 2016. 3

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, 2009. 5

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1, 4

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 1

[5] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 3

[7] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[9] H. Heuer, C. Monz, and A. W. Smeulders. Generating captions without looking beyond objects. *arXiv preprint arXiv:1610.03708*, 2016. 2

[10] D. Hofstadter, D. R. Hofstadter, and E. Sander. *Surfaces and Essences*. Basic Books, 2013. 2

[11] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2

[13] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. *arXiv preprint arXiv:1705.00754*, 2017. 2, 3

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in NIPS*, 2012. 1

[15] G. Lakoff and M. Johnson. *Metaphors we live by*. University of Chicago Press, 1981. 2

[16] A. Lerer, S. Gross, and R. Fergus. Learning physical intuition of block towers by example. In *ICML*, 2016. 3

[17] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 2

[18] V. Michalski, R. Memisevic, and K. Konda. Modeling deep temporal dependencies with recurrent grammar cells. In *Advances in NIPS*, 2014. 3

[19] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in NIPS*, 2016. 1

[20] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta. The curious robot: Learning visual representations via physical interactions. In *ECCV*, 2016. 3

[21] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 3

[22] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 2013. 3

[23] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 2

[24] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015. 2

[25] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012. 3

[26] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004. 2

[27] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, 2014. 1

[28] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 3, 4

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[31] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015. 2

[32] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 6

[33] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 8

[34] Wikipedia. Affordance — wikipedia, the free encyclopedia, 2016. [Online; accessed 9-September-2016]. 1

[35] J. Wu, J. J. Lim, H. Zhang, J. B. Tenenbaum, and W. T. Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *British Machine Vision Conference*, 2016. 2, 3

[36] Z. Wu, Y. Fu, Y.-G. Jiang, and L. Sigal. Harnessing object and scene semantics for large-scale video understanding. In *CVPR*, 2016. 2

[37] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2

[38] L. Yao, N. Ballas, K. Cho, J. R. Smith, and Y. Bengio. Oracle performance for visual captioning. *arXiv preprint arXiv:1511.04590*, 2015. 2

[39] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *CVPR*, 2016. 5