Much Ado About Time: Exhaustive Annotation of Temporal Data

Gunnar A. Sigurdsson[1], Olga Russakovsky[1], Ali Farhadi[3,4], Ivan Laptev[2], and Abhinav Gupta[1,4]

[1]Carnegie Mellon University [2]Inria [3]University of Washington [4]The Allen Institute for AI

http://allenai.org/plato/charades/

Abstract

Large-scale annotated datasets allow AI systems to learn from and build upon the knowledge of the crowd. Many crowdsourcing techniques have been developed for collecting image annotations. These techniques often implicitly rely on the fact that a new input image takes a negligible amount of time to perceive. In contrast, we investigate and determine the most cost-effective way of obtaining high-quality multi-label annotations for temporal data such as videos. Watching even a short 30-second video clip requires a significant time investment from a crowd worker; thus, requesting multiple annotations following a single viewing is an important cost-saving strategy. But how many questions should we ask per video? We conclude that the optimal strategy is to ask *as many questions as possible* in a HIT (up to 52 binary questions after watching a 30-second video clip in our experiments). We demonstrate that while workers may not correctly answer all questions, the cost-benefit analysis nevertheless favors consensus from multiple such cheap-yet-imperfect iterations over more complex alternatives. When compared with a one-question-per-video baseline, our method is able to achieve a 10% improvement in recall (76.7% ours versus 66.7% baseline) at comparable precision (83.8% ours versus 83.0% baseline) in about half the annotation time (3.8 minutes ours compared to 7.1 minutes baseline). We demonstrate the effectiveness of our method by collecting multi-label annotations of 157 human activities on 1,815 videos.

Introduction

Large-scale manually annotated datasets such as ImageNet (Deng et al. 2009) led to revolutionary development in computer vision technology (Krizhevsky, Sutskever, and Hinton 2012). In addition to playing a critical role in advancing computer vision, crowdsourced visual data annotation has inspired many interesting research questions: How many exemplars are necessary for the crowd to learn a new visual concept (Patterson et al. 2015)? How can image annotation be gamified (Von Ahn and Dabbish 2004; von Ahn, Liu, and Blum 2006)? How can we provide richer annotators in the form of visual attributes (Patterson et al. 2014) or object-object interactions (Krishna et al. 2016b)? How can we exhaustively annotate all visual concepts present in an image (Deng et al. 2014)?

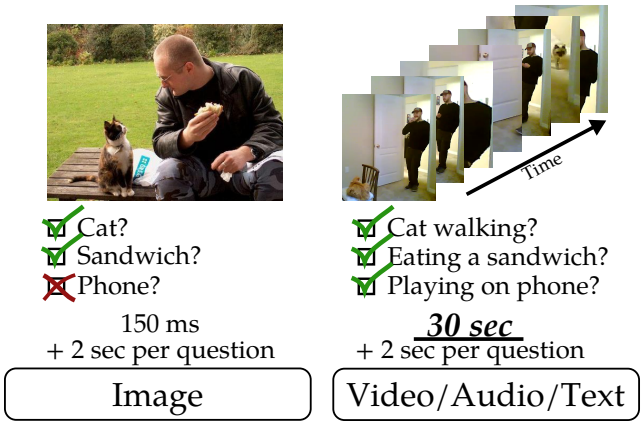

Figure 1: Exhaustively annotating time data has some fundamental differences from image data, and requires different strategies to annotate at scale. In this work we explore the cost-optimal strategies for annotating videos.

Much of the work on visual data annotation has focused on images, but many real-world applications require annotating and understanding *video* rather than image data. A worker can understand an image in only few hundred milliseconds. (Thorpe et al. 1996). Naïvely applying image annotation techniques to data that takes a worker longer to understand, such as data involving time, is prohibitively expensive. Developing effective strategies for temporal annotation is important for multiple domains that require watching, listening, or reading: musical attributes or emotion on songs (Li and Ogihara 2003), news article topics (Schapire and Singer 2000), sentiment analysis (Turney 2002), web page categorization (Ueda and Saito 2002), and video activity recognition (Karpathy et al. 2014).

In this work, we are interested in the following annotation task illustrated in Figure 1: given a video and a set of visual concepts (such as a set of objects or human actions or interesting events), label whether these concepts are present or absent in the video. Efforts such as Glance (Lasecki et al. 2014) focus on quickly answering a question about a video by parallelizing the work across the crowd workforce in 30-second video clips. They are able to get results in near real-time, allowing for *interactive* video annotation. In contrast,

we are interested in annotating a large-scale video dataset where multiple questions (known apriori) need to be answered about each video. Even for a short 30-second video clip, it takes at least 15 seconds at double speed for an annotator to watch the video; thus, asking only a single question at a time is highly inefficient. Efforts such as (Deng et al. 2009; Bragg, Weld, and others 2013) explore multi-label annotation of images but cannot be directly applied to temporal video data because of this inefficiency.

We thus ask: how many questions should we ask workers when annotating a video? We know from psychology research that only on the order of 7 concepts can be kept in short-term memory at a time (Miller 1956). However, our results demonstrate asking many more questions at a time in a single Human Intelligence Task (HIT) can be significantly more efficient. In particular, we demonstrate that asking as many questions as possible, up to 52 questions at a time about a 30-second video in our experiments, provides an optimal tradeoff between accuracy and cost. When compared with a one-question-at-a-time baseline, our method achieves a 10% improvement in recall (76.7% ours versus 66.7% baseline) at comparable precision (83.8% ours versus 83.0% baseline) in about half the annotation time (3.8 minutes ours compared to 7.1 minutes baseline). We empirically verify that our conclusions hold for videos of multiple lengths, explore several strategies for reducing the cognitive load on the workers in the context of video annotation and demonstrate the effectiveness of our method by annotating a new video dataset which can be used for computer vision research on human action detection.

Related Work

Video annotation applications. Video understanding is important for many applications ranging from behavior studies (Coan and Allen 2007) to surveillance (Salisbury, Stein, and Ramchurn 2015) to autonomous driving (Geiger, Lenz, and Urtasun 2012). Large-scale annotated computer vision video datasets (Gorban et al. 2015; Soomro, Roshan Zamir, and Shah 2012; Kuehne et al. 2011; Caba Heilbron et al. 2015; Yeung et al. 2015) enable the development of algorithms that are able to automatically process video collections. However, the lack of large-scale multi-label video datasets makes it difficult to study the intricate interactions between objects and actions in the videos rather than focusing on recognition of one or a handful of concepts.

Efficient video annotation. The key challenge in efficiently annotating video is that it takes a significant time investment. Determining the absence of a concept in an image takes on the order of seconds; in contrast, determining the absence of a concept in a video takes time proportional to the length of the video. On the plus side, there is a lot of temporal redundancy between subsequent video frames, allowing for obtaining annotations only on key frames and interpolating in between. Efforts such as (Yuen et al. 2009; Vondrick, Patterson, and Ramanan 2013; Vijayanarasimhan and Grauman 2012) exploit temporal redundancy and present cost-effective video annotation frameworks. The approaches of (Vondrick and Ramanan 2011; Vijayanarasimhan and Grauman 2012; Fathi et al. 2011) and others additionally incorporate active learning, where the annotation interfaces learns to query frames that, if annotated, would produce the largest expected change in the estimated object track. However, these methods combine human annotation with automatic computer vision techniques, which causes several problems: (1) these techniques are difficult to apply to challenging tasks such as activity recognition where computer vision models lag far behind human ability; (2) these methods are difficult to apply to scenarios where very short or rare events, such as shoplifting, may be the most crucial, and (3) the resulting hybrid annotations provide a biased testbed for new algorithms.

Glance (Lasecki et al. 2014) focuses on parallelizing video annotation effort and getting an answer to a single question in real-time, but does not explore exhaustive video annotation where multiple questions need to be answered. Our work can be effectively combined with theirs: they parallelize annotation in 30-second video chunks, while we explore the most effective ways to obtain multiple labels simultaneously for every 30-second video.

Action recognition datasets. Some existing large-scale action datasets such as EventNet (Ye et al. 2015) or Sports-1M (Karpathy et al. 2014) rely on web tags to provide noisy video-level labels; others like THUMOS (Gorban et al. 2015) or MultiTHUMOS (Yeung et al. 2015) employ professional annotators rather than crowdsourcing to label the temporal extent of actions.

There are two recent large-scale video annotation efforts that successfully utilize crowdsourcing. The first effort is ActivityNet (Heilbron and Niebles 2014) which uses a proposal/verification framework similar to that of ImageNet (Deng et al. 2009). They define a target set of actions, query video search engines for proposal videos of those actions and then ask crowd workers to clean up the results. The second effort (Sigurdsson et al. 2016) entirely crowdsources the creation of a video dataset: one worker writes a video script containing a few target objects/actions, another one acts out the script and films the video, and others verify the work. In both these efforts, each video comes pre-associated with one or a handful of action labels, and workers are tasked with verifying these labels. In contrast, we're interested in the much more challenging problem of multi-label video annotation beyond the provided labels.

Multi-label image annotation. Increasingly more complex image annotations are provided in recent dataset (Bigham et al. 2010; Lin et al. 2014; Krishna et al. 2016b). Multi-label image annotation has been studied by e.g., (Von Ahn and Dabbish 2004; Deng et al. 2014; Bragg, Weld, and others 2013; Zhong et al. 2015; Noronha et al. 2011). We incorporate insights from these works into our video annotation framework. We use a hierarchy of concepts to accelerate multi-label annotation following (Deng et al. 2014; Bragg, Weld, and others 2013). Inspired by (Krishna et al. 2016a), we explore using cheap but error-prone annotation interfaces over thorough but more expensive formulations.

Method for multi-label video annotation

We are given a collection of M videos and a set of N target labels: for example, a list of target object classes, e.g., "cat," "table," or "tree," or a list of human actions, e.g., "reading a book" or "running." The goal is to obtain $M \times N$ binary labels, corresponding to the presence or absence of each of the N target concepts in each of the M videos. These labels can then be used for a variety of applications from training computer vision models (Karpathy et al. 2014) to studying human behavior (Coan and Allen 2007).

We are particularly interested in situations where the label space N is large: $N = 157$ in our experiments. As a result, the key challenge is that workers are not able to remember all N questions at the same time; however every time a worker is required to watch a video of length L during annotation, they have to invest an additional L seconds of annotation time. We focus on video annotation in this work but our findings may be applicable to any media (e.g., audio, text) where a non-trivial amount of time L is required to process each input.

Multiple question strategy

Our strategy is to ask all N target questions at the same time about each video, even if N is much higher than the 7 concepts that people can commit to short-term memory (Miller 1956). We randomize the order of questions and ask workers to select only the concepts that occur within the video. This naturally leads to lower recall r than if we ask only a handful of questions that the workers would be more likely to read carefully. However, there are two advantages.

Advantage #1: Low annotation times. Since only one worker has to watch the video instead of asking N different workers to watch the video and annotate one label each, this recall r is obtained with relatively little time investment t. This makes it a highly effective strategy combined with consensus among multiple workers (Sheshadri and Lease 2013). Given a fixed time budget T, we can repeat the annotation process $\frac{T}{t}$ times with different workers. Assume the workers are independent and we count the concept as present in the image if at least one worker annotates it. Then our expected recall in T time is

$$\text{ExpectedRecall} = 1 - (1 - r)^{\frac{T}{t}} \quad (1)$$

since each worker will miss a concept with $1 - r$ probability, and a concept won't be annotated only if all $\frac{T}{t}$ workers independently miss it.

Advantage #2: High precision. The $M \times N$ label matrix is naturally sparse since most concepts do not occur in most videos. When workers are faced with only a small handful of concepts and none of them occur in the video, they get nervous that they are not doing the task correctly. Then, they are more likely to provide some erroneous positive labels. However, when the workers are faced with many concepts at the same time and asked to select the ones that occur in the video, they find the task much more enjoyable. They get satisfaction out of being able to spot several target concepts that occur in the video and are less likely to erroneously select additional concepts.

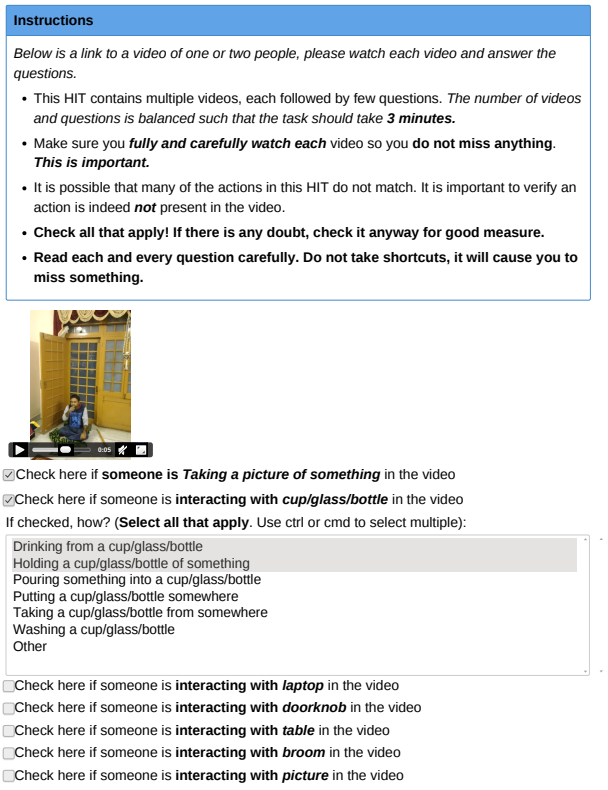

Figure 2: Our multi-question video annotation interface.

Practical considerations

In designing an effective multi-question video annotation interface shown in Figure 2, we incorporate insights from image annotation (Deng et al. 2014) to reduce the space of N labels and from video annotation (Lasecki et al. 2014) to compress the video length L.

Semantic hierarchy. Following (Deng et al. 2014) we create a semantic hierarchical grouping of concepts to simplify the multi-label annotation. However, (Deng et al. 2014) use the hierarchy differently. They ask one question at a time about a matrix of images, e.g., "click on all images which contain an animal." They then ask a low-level question, e.g., "click on all images which contain a dog," on a smaller matrix of images which were positive for the prior question. In contrast, we use the concept hierarchy similar to (Xiao et al. 2014) to simplify our annotation interface on a single video.

Playback speed. Videos of average length of 30 seconds are played at 2x speed following (Lasecki et al. 2014). In this way, worker time is not unnecessarily wasted but they are able to perceive and accurately annotate the target concepts.

Instructions. Workers are instructed to carefully watch each video to not miss anything, and check all concepts that apply. Since most concepts do not occur in the video, workers are asked to only check the boxes for the concepts that do occur and to ignore the others. We verify this design choice in the experiments below.

Experiments

We begin by describing the setup used to evaluate our method, including steps taken to control for factors of variation across different crowdsourcing experiments. We then present a series of smaller-scale experiments on 100-150 videos at a time investigating (1) the effects of varying the number of questions in the annotation interface, and (2) the effectiveness of strategies for reducing cognitive load on workers during annotation. We conclude by bringing our findings together and evaluating our large-scale multi-label video annotation pipeline and the resulting dataset.

Data and evaluation setup

We use a recent large-scale video dataset (Sigurdsson et al. 2016) with a focus on common household activities. We use a subset of 1,815 videos, on average 30.1 seconds long. The target labels are 157 activity classes such as *Someone is running* and *Putting a cup somewhere* provided with the dataset. The videos are associated with some labels apriori, similar to ImageNet (Deng et al. 2009) and ActivityNet (Caba Heilbron et al. 2015). Figure 3 shows some examples. This misses additional activities also present in the video, making it difficult to evaluate computer vision algorithms and to study interactions between different actions. We demonstrate how to cost-effectively collect exhaustive annotations for this dataset. The annotations have been released along with the dataset.

Evaluating recall. We use the originally provided action labels to evaluate the recall of our multi-label annotation algorithms. There were on average 3.7 activities labeled per video in this dataset. The activities follow a long-tailed distribution: some occur in as many as 1391 videos, others in only 33. Each activity occurs in 42 videos on average.

Evaluating precision. Precision is more difficult to evaluate since to the best of our knowledge no large-scale video dataset is annotated with hundreds of visual concepts. Annotating the videos in this dataset exhaustively in a straightforward way is prohibitively expensive, which is exactly what we are trying to address in this work. We adopt a middle ground. After obtaining a set of candidate labels from the annotators, we perform a secondary verification step. In the verification task, workers have to annotate the temporal extent of the action in the video or specify it is not present in the video. This serves as an evaluation of the precision of multi-label annotation. In addition, this provides temporal action annotations which we will also publicly release.

Semantic hierarchy. The 157 target human activities are grouped based on the object being interacted with to simplify the annotation interface. The annotator first sees several questions such as "Check here if someone is interacting with a book in the video" or "Check here if someone is interacting with shoes in the video." If the annotator says *yes* someone is interacting with a book, s/he will be asked to select one or more of the types of interaction: closing a book? opening a book? holding a book? putting a book somewhere?

We create 33 object groups, each group with 4.2 activities on average. Additionally, 19 activities (such as *Someone is laughing*, *Someone is running somewhere*) do not belong to any group and are asked individually. Thus, we obtain 52 high-level questions which cover all of the label space; the exact hierarchy is provided in the Appendix.

Figure 3: Examples from the video dataset of (Sigurdsson et al. 2016). The videos contain complex human activities that require the annotator to carefully watch each video.

Crowdsourcing setup

During the study, 674 workers were recruited to finish 6,337 tasks on Amazon Mechanical Turk. We summarize some key crowdsourcing design decisions here.

Quality control. Workers were restricted to United States, with at least 98% approval rate from at least 1000 tasks. We used recall, annotation time, and positive rate to flag outliers, which were manually examined and put on a blacklist. To maintain a good standing with the community all work completed without clear malice was approved, but bad workers were prohibited from accepting further work of this type.

In Figure 4 the relationship between how much time an individual worker spends on a task and quality of the annotation is presented. We can see that apart from clear outliers, there is no significant difference, and in this work we treat the worker population as following the same distribution, and focus on the time difference between different methods.

Uncontrolled factors. There are many sources of variation in human studies, such as worker experience (we observed worker quality increasing as they became more familiar with our tasks) or time of day (full-time workers might primarily be available during normal business hours). We attempted to minimize such variance by deploying all candidate methods at the same time within each experiment.

Payment. In order to verify our hypothesis that it is best to ask multiple questions about a video simultaneously, we need to evaluate interfaces with a varying number of questions per video. However, we want to maintain as much consistency as possible outside of the factor we're studying. We use a single type of Human Intelligence Task (HIT) where workers are provided with V videos and Q questions for

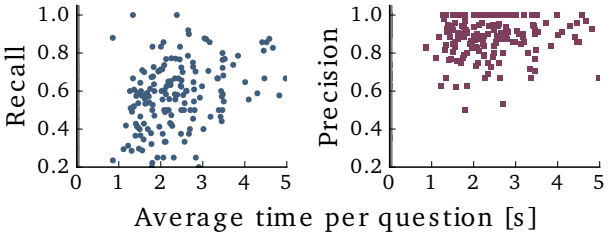

Figure 4: Workers that spend more time answering questions have marginally higher accuracy (Pearson's correlation of time with recall is 0.227 and with precision is 0.036). However this trend is so slight that we ignore it and instead focus on improving the annotation workflow as a whole.

each video using the interface of Figure 2. When we increase the number of questions Q per video, we decrease the number of videos V, and vice versa, to keep the expected annotation effort consistent within the HIT.

To do this, we ran some preliminary experiments and analyzed the average amount of time it takes to label a video in our dataset with Q questions. Figure 5 shows the relationship between number of questions Q and time. The least-squares line of best fit to this data is

$$T = 14.1 + 1.15Q \quad (2)$$

Thus it takes an average of 14.1 seconds to watch a video and an additional 1.15 seconds to answer each question. This is consistent with our expectations: an average video in our dataset is 30.1 seconds long, which we play at double speed, and binary questions take on the order of 1-2 seconds to answer (Krishna et al. 2016b).

We varied the number of videos in each HIT using Equation 2 to target about 150 seconds of expected annotation effort. We paid \$0.40 per HIT, amounting to about \$9.60 per hour.

Multiple question interface. We report results on annotating the 157 activities using the 52-question semantic hierarchy.[1] Our method solicits labels for all 52 questions and corresponding sub-questions in the same interface as shown in Figure 2. When evaluating interfaces with a smaller number of questions k, we partition the 52 questions into $\frac{52}{k}$ subsets randomly. Multiple workers then annotate the video across $\frac{52}{k}$ tasks, and we accumulate the results.[2] An *iteration* of annotation refers to a complete pass over the 52 questions for each video. We can then directly compare the annotations resulting from interfaces with different values of k.

Effect of varying the number of questions

So far we described the data, the evaluation metrics and the crowdsourcing setup. We are now ready to begin experimenting with different annotation strategies.

[1] We additionally verified that all conclusions hold if we are interested in only the 52 high-level activities as well.

[2] Some of the questions take longer than others, and thus some subsets may take longer to annotate than others. However, we report cumulative results after all subsets have been annotated and thus the variations in time between the subsets is irrelevant.

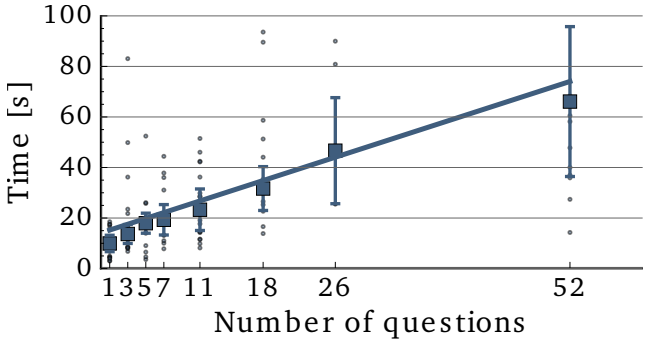

Figure 5: The relationship between number of questions in the interface and the amount of time it takes. We use it to maintain a consistent amount of annotation effort across HITs while varying the number of questions in the interface.

We begin by varying the number of questions the workers are asked after watching each video: from only 1 question per video (very time-inefficient since 52 workers have to independently watch the video) up to all 52 questions at the same time (potentially daunting for the workers). We run the annotation experiment on 140 videos, and report the time, recall and precision after one iteration of annotation, i.e., after workers have had a chance to answer all 52 questions about each video using the different interfaces.

Advantages of asking multiple questions. There are two advantages to asking multiple questions together rather than one-at-a-time, as shown in Figure 6. The first advantage is *low annotation time*: the time for one iteration of annotation drastically decreases as a function of the number of questions. Concretely, it takes 8.61 minutes per video with the 1-question interface versus only 1.10 minutes per video with the 52-question interface. This is expected, as the time to watch the video gets amortized.

The second advantage to asking multiple questions together is *increased precision* of annotation. Concretely, while annotation precision is only 81.0% with the 1-question interface, it rises up to 86.4% with the multiple 52-question interface. When only one question per video is asked, almost certainly all answers in a HIT will be negative, since only a small subset of the target activities occur in each 30-second video. Workers have reported being concerned when all answers appear negative. We hypothesize that as a result, they may erroneously answer positively if they have any suspicions about the activity being present, which decreases the precision of annotation in the 1-question interface.

Drawback of asking multiple questions. The one drawback of asking multiple questions is *decreased recall*. When asked only one question per video, workers are able to achieve 56.3% recall, whereas when asked all 52 questions at once they only correctly identify labels with 45.0% recall. This is because it is challenging to keep 52 questions in memory while watching the video or the entire video in memory while answering the questions. Interestingly, Fig-

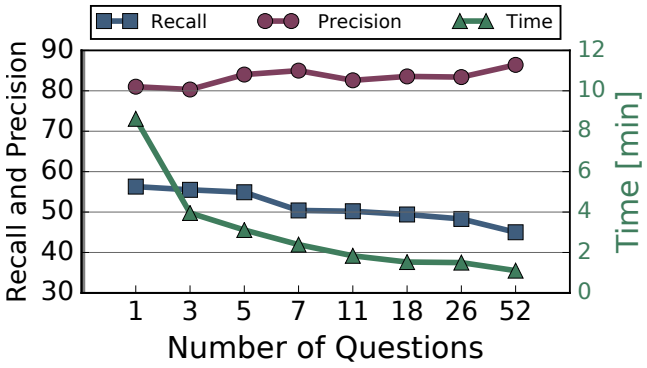

Figure 6: Accuracy (*left axis*) and time (*right axis*) of annotation as a function of the number of questions in the interface (*x-axis*). While recall is higher with fewer questions, this is at the cost of significantly higher annotation time.

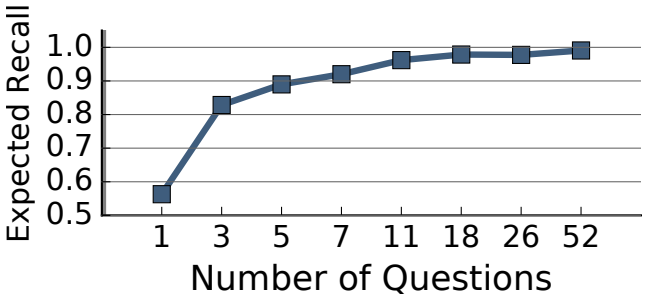

Figure 7: Expected recall given a fixed time budget (simulated using Equation 2) for interfaces with a varying number of questions. The budget is 8.61 minutes per video, enough to run 1 iteration of annotation with the 1-question interface.

ure 6 shows a sharp drop in recall beyond 5-7 questions in the interface, which is the number of concepts people can keep in short-term memory (Miller 1956).

Fixing the drawback. Even though recall is lower when asking multiple questions about a video, it is obtained in significantly less annotation time. Given a fixed time budget, we can compute the expected recall if we were to ask multiple workers to do the annotation by referring back to Equation 1. In particular, assume we are given 8.61 minutes that it takes to fully annotate a video using the 1-question interface. In this amount of time, we can ask at least 7 workers to annotate it with the 52-question interface (since it only takes 1.10 minutes per iteration). Figure 7 reports the expected recall achievable in 8.61 minutes using the different interfaces. We conclude that the many-question interfaces are better than the few-question interfaces not only in terms of time and precision, but also in terms of recall for a fixed time budget. We will revisit this in later experiments.

Worker behavior. Besides quantitatively evaluating the different interfaces according to the standard metrics, it is also informative to briefly look into annotator behavior.

Figure 8 reports the number of interactions of workers

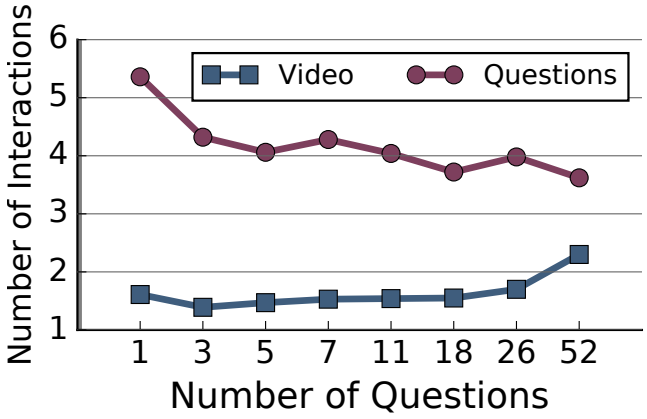

Figure 8: The number of times workers paused or synced the video (*video*) and the number of questions answered affirmatively after an iteration of annotation (*questions*) as a function of the number of questions in the interface.

with the video: i.e., the number of times they pause or seek the video. We observe that the interactions with the video generally increase with the question count, suggesting that workers may be watching the video more carefully when asked more questions. Interestingly, however, with only a single question the users seem to hurry through the video.

Figure 8 additionally reports the average number of questions answered affirmatively by the workers after an iteration of annotation. As the number of questions in the interface increases, the average number of affirmative answers after 52 questions have been answered decreases from 5.36 to 3.62. We hypothesize that when multiple questions are presented to the workers simultaneously, they feel satisfied once they are able to answer a handful of them positively; when faced with only a small number of questions, they feel increased pressure to find more positive answers. This contributes to both the increase in recall and the drop in precision.

Worker feedback. Finally, we asked workers to report their enjoyment of the task on a scale of 1 (lowest) to 7 (highest). Average enjoyment ranged from 5.0 to 5.3 across the different interfaces, indicating that workers were equally pleased with both few-question and many-question tasks.[3]

Targeting the UI for different number of questions

So far we investigated the effect that number of questions have on the accuracy and efficiency of annotation, while keeping all other factors constant. However, using the same user interface and annotation workflow for different numbers of questions may not be optimal. For example workers tend to worry when asked too many negative questions in a row in an interface with a few questions, or may not read all questions in detail in an interface with many questions.

In this section, we use the 3-question interface for the *few-questions* setting, and the 26-question interface for the

[3]In our preliminary experiments we did not use Equation 2 to control for the amount of work within each HIT; worker enjoyment was then strongly inversely correlated with the amount of work.

many-questions setting. We run a series of experiments investigating strategies for improving the UI. We discover two strategies for improving the few-questions interface and conclude that our many-questions interface is optimal.

Positive bias. When using the few-questions interface, most answers within a HIT are expected to be negative since most target activities are not present in the videos. This has two undesirable effects: (1) workers may start paying less attention, and (2) workers may get nervous and provide erroneous positive answers, lowering the annotation precision.

To overcome this, we duplicate questions known to be positive and inject them such that approximately 33% of the questions are expected to be positive. This forces the workers to pay closer attention and be more active in the annotation; on the downside, this increases the number of questions per annotation from 52 to 78 including the duplicates.

In an experiment on 150 videos, injecting such positive bias into the few-questions interface improves on all three metrics: recall, precision and time of annotation. Recall increases from 53.2% to 57.9% with positive bias,[4] precision increases slightly from 79.0% to 81.3% with positive bias, and time for an iteration of annotation drops from 4.6 minutes to 3.6 minutes, likely because workers trust their work more and thus are able to annotate faster. Workers also report slightly higher enjoyment: on a scale of 1 (lowest) to 7 (highest), they report 5.8 enjoyment of the task with positive bias versus 5.5 without. We incorporate positive bias into the few-question interface in future experiments.

Grouping. Prior work such as (Deng et al. 2009) has demonstrated that asking about the same visual concepts across multiple images reduces the cognitive load on workers and increases annotation accuracy. In our second experiment, we apply the same intuition to videos: we randomly group questions together and make sure that all questions are the same for all videos within a single HIT. Residual question not part of the groups, and groups too small to fill a whole task were discarded, but each question was presented both in the context of grouping and not, for a fair comparison.

In the few-questions interface, grouping improves the precision and the time of annotation, albeit at a slight reduction in recall. Specifically, in an experiment on 100 videos, precision increases from 77.7% to 81.4% when grouping is added. Annotation time per iteration drops from 5.9 minutes to 5.1 minutes with grouping; however, recall also drops from 70.4% to 67.2% with grouping. To determine if the drop in recall is a concern, we refer back to Equation 1 to compute the expected recall for a fixed time budget. In 5.9 minutes (enough for one iteration without grouping), we expect a recall of 72.3% with grouping, higher than 70.4% recall without. Thus, we conclude that grouping is strictly beneficial in the few-question setting as hypothesized, and we use it in future experiments.

We also investigated the effect of grouping in the many-question interface, but concluded it's unhelpful. Recall with grouping is 55.2%, much lower than 62.0% without grouping. Even though annotation time is faster (1.4 minutes per iteration with grouping compared to 1.6 minutes per iteration without), this is not enough to compensate for the drop in recall: the expected recall given a budget of 1.6 minutes of annotation is still only 61.2% with grouping compared to 62.0% without. Further, precision is also lower with grouping: 79.0% with grouping compared to 80.2% without. We hypothesize that this is because workers are not able to remember all 26 questions anyway, so grouping only provides a false sense of security (as evidenced by the speedup in annotation time). We do not use grouping in the multi-question interface in future experiments.

Video summary. Having discovered two strategies for improving the few-question interface (positive bias and grouping), we turn our attention to strategies targetting the multi-question setup. The main challenge in this setting is that workers may be overwhelmed by the number of questions and may not read them all carefully.

To better simulate a scenario where the worker has to pay careful attention to the video, we add an additional prompt to the many-questions interface. In an experiment on 100 videos, workers were asked to "please describe with approximately 20 words what the person/people are doing in the video." This adds on average 36 seconds per iteration, yielding 2.1 minutes of annotation time with the additional prompt versus 1.5 without. However, the extra time does not translate to noticeable benefits in annotation accuracy: recall drops slightly to 53.2% with the prompt compared to 54.2% without, although precision increases slightly to 88.3% with the prompt compared to 87.1% without. We conclude that adding the prompt has no significant impact on the accuracy of annotation despite a 1.4x increase in annotation time.

Forced responses. The final investigation into improving the many-questions interface is asking workers to actively select a yes/no response to every question rather than simply checking a box only if an action is present. Intuitively this forces the workers to pay attention to every question. However, this again produces no improvements in accuracy, indicating that workers are already working hard to provide the most accurate responses and are only confused by the additional forced responses.

Concretely, we experimented on 100 videos and observed a drop in recall to 55.7% with the forced responses compared to 63.3% without as well as a drop in precision to 84.6% with forced responses compared to 88.8% without. Further, annotation time increases to 2.2 minutes per video with forced responses versus 1.6 minutes without. Thus forcing workers to read every question is in fact appears harmful: it is better for them to focus on watching the video and only skim the questions.

Conclusions. We thoroughly examined the annotation interface in the few-questions and many-questions setting. We discover that positive bias and grouping are effective strategies for improving the few-questions UI, and incorporate them in future experiments. For the many-questions setting, simply randomizing the questions and allowing the workers

[4] To maintain a fair comparison, answers to duplicate questions are ignored during evaluation. Thus the time it takes to answer them is also ignored when computing annotation time per iteration.

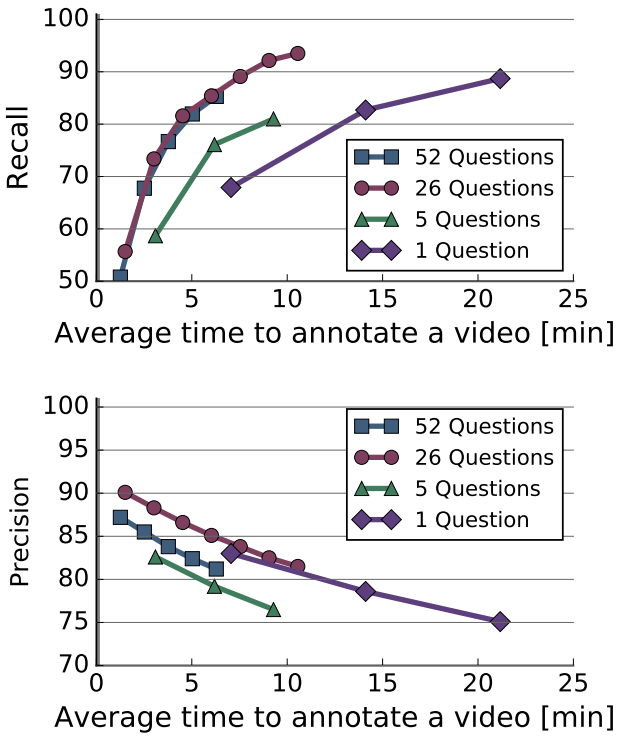

Figure 9: Recall (*top*) and precision (*bottom*) with multiple iterations of annotation. Each square represents one iteration. We can see that since each annotation iteration with the 52-question interface is much cheaper, it quickly matches the performance of the more time-costly alternatives.

to select the actions that appear in the video is shown to be more effective than any other baseline.

Multi-iteration video annotation

So far we established that (1) the many-questions interface provides a more effective accuracy to annotation cost tradeoff on expectation than the few-questions interface when all other factors are kept the same, (2) the few-questions interface can be further improved by the addition of positive bias and grouping, and (3) the many-questions interface we proposed is optimal as is. In this section we bring all these findings together and conclusively demonstrate that our many-question annotation strategy is strictly better than the few-questions alternatives for practical video annotation.

Advantages of asking multiple questions. In previous sections we computed the expected recall across multiple iterations of annotations for a fixed time budget to compare different methods; here, we report the results in practice. We run multiple iterations of annotation and consider a label positive if at least one worker marks it as such. Thus, recall steadily increases with the number of iterations while precision may drop as more false positives are added.

Figure 9 reports recall and precision as a function of annotation time. For the few-question interfaces (5-questions and 1-question) we include the positive bias and grouping strategies which have been found helpful. Nevertheless, we observe a clear advantage of the multi-question methods over other alternatives.

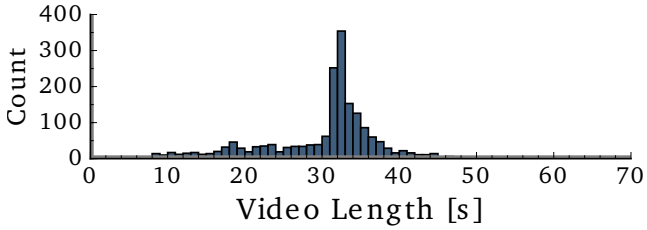

Figure 10: Statistics from the dataset. Histogram of the lengths of the videos, where we can see that the videos have various lengths enabling analysis based on content length.

For example, given 7.1 minutes required to annotate a video with the 1-question interface, we are able to run two iterations with the 5-question interface (taking up 6.2 minutes), and five iterations with 52-questions (taking up 6.3 minutes). With this annotation budget, the 52-question interface obtains a recall of 85.3%, which is 10.5% higher than the 74.8% recall with 5-questions and 18.6% higher than the 66.7% recall with 1-question. Further, the 52-question interface obtains precision of 81.2%, which is 6.6% higher than the 74.6% precision with 5-questions and slightly lower by 1.8% than the 83.0% precision with the 1-question interface.

In another example, in about half the annotation time (3.8 versus 7.1 minutes) we achieve a 10% improvement in recall (76.7% with three iterations of 52-questions versus 66.7% with one iteration of 1-question) at comparable precision (83.8% with 52-questions versus 83.0% with 1-question).

We conclude that simultaneously asking multiple questions per video, as many as 26 or even 52, is significantly more time-effective than asking only a handful of questions. When comparing the 26-question with the 52-question interface in Figure 9, the results are remarkably similar: recall per unit time is almost identical, although precision is slightly higher for 26-questions. We conclude that asking more questions per video is not harmful to annotation quality; we further verify this below by evaluating on videos of different length.

Effect of video length. We investigate whether these conclusions hold for different video lengths – for example, an image is just a zero-length video, so would our conclusions still apply? Our dataset contains videos of varying length as shown in Figure 10 and we group the videos into three groups: 0-20 seconds, 20-40 seconds and 40-60 seconds long.

Figure 11 reports the recall of the different methods for each of the three groups, following the same experimental design as before. For shorter videos that require little time to process, the exact annotation interfaces make little difference. This suggests that in the case of images our method would be as effective as the standard one-question baseline.

Importantly, as the content gets longer the benefit of our method becomes more pronounced. For example, on 40-60

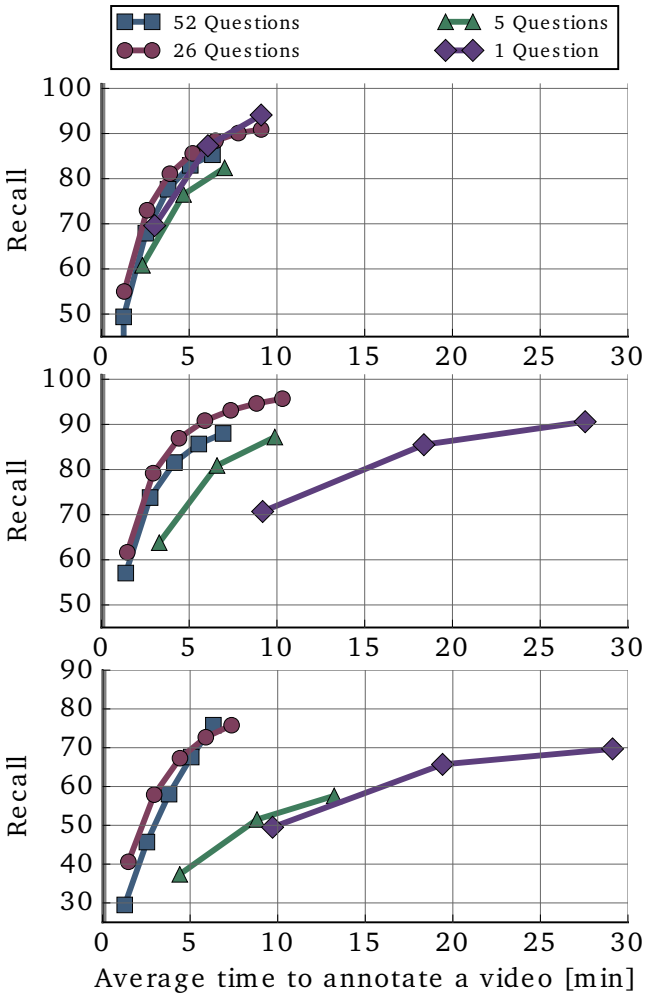

Figure 11: Breakdown of Figure 9 for different video lengths: *(top)* 0-20 second videos, *(middle)* 20-40 second videos, *(bottom)* 40-60 second videos. The benefit of the many-question interfaces is more prominent with increased content length.

second videos for a fixed annotation budget of 4.4 minutes (enough to run one iteration with the 5-question interface), our 52-question method achieves 62.7% recall compared to only 37.4% with the 5-question baseline (a 25.3% improvement!) and 83.1% precision compared to only 79.4% precision of the 5-question baseline.

Annotated dataset

We used our annotation strategy to collect additional annotations for the video dataset of (Sigurdsson et al. 2016). This amounted to 443,890 questions answered, resulting in 1,310,014 annotations for the 1,815 videos. This increased the density of annotation on the dataset from 3.7 labels per video on average (which were available apriori based on the data collection procedure) to 9.0 labels per video. In addition, when evaluating the precision of annotation we also collected temporal annotation of *when* the actions took place

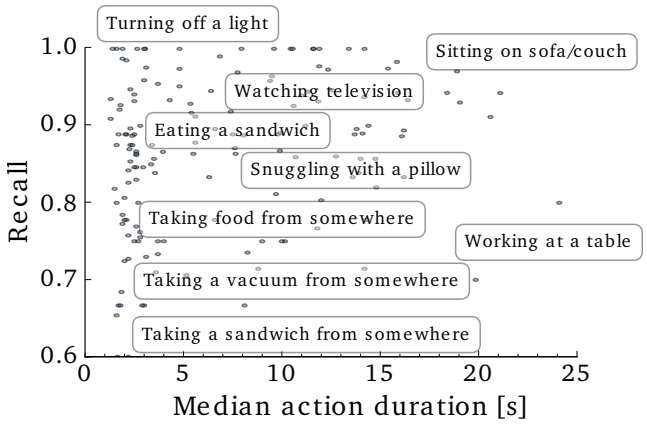

Figure 12: Annotation recall (*y-axis*) as a function of the average duration in the video (*x-axis*) for every one of the target 157 actions. Our method for multi-label video annotation is effective for labeling both long- and short-duration events.

in the video. This yielded 66,963 action instances. We will release all the annotations to enable future research in both crowdsourcing and in computer vision.

Using these temporal annotations, we verify that using our method we are able to successfully annotate both actions that are long and short in the video. For every one of the 157 target actions, we compute the average (median) length of its instances in the videos as well as the recall of our annotations. Figure 12 plots recall as a function of action duration. As expected, recall tends to be slightly higher for actions that are longer in the video but not significantly (Pearson correlation of 0.178). We conclude our method is effective at annotating both long and short events.

Discussion & Conclusions

We explored the challenging problem of multi-label video annotation. In contrast to insights obtained from studying crowdsourcing of video annotation, we demonstrated that asking multiple questions simultaneously about a video provides the most effective tradeoff between annotation time and accuracy. While we observed that accuracy decreases with additional questions for each video, this drop was not sufficient to warrant the significant cost of only a few questions per video. Furthermore, we observed that the performance gap between cheap fast methods over slow careful methods grows with increasing content length. In conclusion, our results suggest that optimal strategy of annotating data involving time is to minimize the cost in each iteration through sufficiently many questions, and simply run multiple iterations of annotation.

Acknowledgments

This work was partly supported by ONR MURI N00014-16-1-2007, ONR N00014-13-1-0720, NSF IIS-1338054, ERC award ACTIVIA, Allen Distinguished Investigator Award, gifts from Google, and the Allen Institute for Artificial Intelligence. The authors would like to thank Gül Varol for help-

ful discussions. Finally, the authors want to extend thanks to all the workers at Amazon Mechanical Turk.

References

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; et al. 2010. VizWiz: nearly real-time answers to visual questions. In *User Interface Software and Technology (UIST)*, 333–342.

Bragg, J.; Weld, D. S.; et al. 2013. Crowdsourcing multi-label classification for taxonomy creation. In *Human Computation and Crowdsourcing (HCOMP)*.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *Computer Vision and Pattern Recognition (CVPR)*.

Coan, J. A., and Allen, J. J. B. 2007. *Handbook of Emotion Elicitation and Assessment*. New York: Oxford University Press.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*.

Deng, J.; Russakovsky, O.; Krause, J.; Bernstein, M. S.; Berg, A.; and Fei-Fei, L. 2014. Scalable multi-label annotation. In *SIGCHI Conference on Human Factors in Computing Systems*.

Fathi, A.; Balcan, M.; Ren, X.; and Rehg, J. 2011. Combining self training and active learning for video segmentation. In *British Machine Vision Conference (BMVC)*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*.

Gorban, A.; Idrees, H.; Jiang, Y.-G.; Roshan Zamir, A.; and Laptev. 2015. THUMOS challenge: Action recognition with a large number of classes. `http://www.thumos.info/`.

Heilbron, F. C., and Niebles, J. C. 2014. Collecting and annotating human activities in web videos. In *Proceedings of International Conference on Multimedia Retrieval*, 377. ACM.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*.

Krishna, R.; Hata, K.; Chen, S.; Kravitz, J.; Shamma, D. A.; Fei-Fei, L.; and Bernstein, M. S. 2016a. Embracing error to enable rapid crowdsourcing. In *SIGCHI Conference on Human Factors in Computing Systems*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; Bernstein, M.; and Fei-Fei, L. 2016b. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR* abs/1602.07332.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*, 2556–2563. IEEE.

Lasecki, W. S.; Gordon, M.; Koutra, D.; Jung, M. F.; Dow, S. P.; and Bigham, J. P. 2014. Glance: Rapidly coding behavioral video with the crowd. In *User Interface Software and Technology (UIST)*, 551–562. ACM.

Li, T., and Ogihara, M. 2003. Detecting emotion in music. In *Proceedings of the fourth international conference on music information retrieval (ICMIR)*, volume 3, 239–240.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollr, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*.

Miller, G. A. 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review* 63(2):81.

Noronha, J.; Hysen, E.; Zhang, H.; and Gajos, K. Z. 2011. Platemate: Crowdsourcing nutritional analysis from food photographs. In *User Interface Software and Technology (UIST)*.

Patterson, G.; Xu, C.; Su, H.; and Hays, J. 2014. The SUN attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision* 108(1-2):59–81.

Patterson, G.; Van Horn, G.; Belongie, S.; Perona, P.; and Hays, J. 2015. Tropel: Crowdsourcing detectors with minimal training. In *Human Computation and Crowdsourcing (HCOMP)*.

Salisbury, E.; Stein, S.; and Ramchurn, S. 2015. Crowdar: augmenting live video with a real-time crowd. In *Human Computation and Crowdsourcing (HCOMP)*.

Schapire, R. E., and Singer, Y. 2000. Boostexter: A boosting-based system for text categorization. *Machine learning* 39(2):135–168.

Sheshadri, A., and Lease, M. 2013. Square: A benchmark for research on computing crowd consensus. In *Human Computation and Crowdsourcing (HCOMP)*.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Laptev, I.; Farhadi, A.; and Gupta, A. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv e-prints*.

Soomro, K.; Roshan Zamir, A.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*.

Thorpe, S.; Fize, D.; Marlot, C.; et al. 1996. Speed of processing in the human visual system. *Nature* 381(6582):520–522.

Turney, P. D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Association for Computational Linguistics (ACL)*.

Ueda, N., and Saito, K. 2002. Parametric mixture models for multi-labeled text. In *Advances of Neural Information Processing Systems (NIPS)*.

Vijayanarasimhan, S., and Grauman, K. 2012. Active frame selection for label propagation in videos. In *European Conference on Computer Vision (ECCV)*.

Von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 319–326. ACM.

von Ahn, L.; Liu, R.; and Blum, M. 2006. Peekaboom: A game for locating objects in images. In *SIGCHI Conference on Human Factors in Computing Systems*.

Vondrick, C., and Ramanan, D. 2011. Video Annotation and Tracking with Active Learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Vondrick, C.; Patterson, D.; and Ramanan, D. 2013. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision* 101(1):184–204.

Xiao, J.; Ehinger, K. A.; Hays, J.; Torralba, A.; and Oliva, A. 2014. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision (IJCV)*.

Ye, G.; Li, Y.; Xu, H.; Liu, D.; and Chang, S.-F. 2015. Eventnet: A large scale structured concept library for complex event detection in video. In *ACM International Conference on Multimedia*.

Yeung, S.; Russakovsky, O.; Jin, N.; Andriluka, M.; Mori, G.; and Li, F.-F. 2015. Every moment counts: Dense detailed labeling of actions in complex videos. *CoRR* abs/1507.05738.

Yuen, J.; Russell, B.; Liu, C.; and Torralba, A. 2009. Labelme video: Building a video database with human annotations. In *International Conference on Computer Vision (ICCV)*.

Zhong, Y.; Lasecki, W. S.; Brady, E.; and Bigham, J. P. 2015. RegionSpeak: Quick comprehensive spatial descriptions of complex images for blind users. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*.

Appendix

Below we present the hierarchy of concepts used in our multi-label annotation interface. The 157 human actions are organized into a hierarchy according to the object the human is interacting with. This hierarchy is used to simplify the interface.

clothes: Holding some clothes, Putting clothes somewhere, Taking some clothes from somewhere, Throwing clothes somewhere, Tidying some clothes.

door: Closing a door, Fixing a door, Opening a door .

table: Putting something on a table, Tidying up a table, Washing a table, Sitting at a table, Working at a table, Sitting on a table.

phone/camera: Holding a phone/camera, Playing with a phone/camera, Putting a phone/camera somewhere, Taking a phone/camera from somewhere, Talking on a phone/camera.

bag: Holding a bag, Opening a bag, Putting a bag somewhere, Taking a bag from somewhere, Throwing a bag somewhere.

book: Closing a book, Holding a book, Opening a book, Putting a book somewhere, Taking a book from somewhere, Throwing a book somewhere, Watching/Reading/Looking at a book.

towel: Holding a towel/s, Putting a towel/s somewhere, Taking a towel/s from somewhere, Throwing a towel/s somewhere, Tidying up a towel/s, Washing something with a towel.

box: Closing a box, Holding a box, Opening a box, Putting a box somewhere, Taking a box from somewhere, Taking something from a box, Throwing a box somewhere.

laptop: Closing a laptop, Holding a laptop, Opening a laptop, Putting a laptop somewhere, Taking a laptop from somewhere, Watching a laptop or something on a laptop, Working/Playing on a laptop.

shoe/shoes: Holding a shoe/shoes, Putting shoes somewhere, Putting on shoe/shoes, Taking shoes from somewhere, Taking off some shoes, Throwing shoes somewhere.

chair: Sitting in a chair, Standing on a chair.

food: Holding some food, Putting some food somewhere, Taking food from somewhere, Throwing food somewhere.

sandwich: Eating a sandwich, Holding a sandwich, Putting a sandwich somewhere, Taking a sandwich from somewhere.

blanket: Holding a blanket, Putting a blanket somewhere, Snuggling with a blanket, Taking a blanket from somewhere, Throwing a blanket somewhere, Tidying up a blanket/s.

pillow: Holding a pillow, Putting a pillow somewhere, Snuggling with a pillow, Taking a pillow from somewhere, Throwing a pillow somewhere.

shelf: Putting something on a shelf, Tidying a shelf or something on a shelf.

picture: Reaching for and grabbing a picture, Holding a picture, Laughing at a picture, Putting a picture somewhere, Watching/looking at a picture.

window: Closing a window, Opening a window, Washing a window, Watching/Looking outside of a window.

mirrow: Holding a mirror, Smiling in a mirror, Washing a mirror, Watching something/someone/themselves in a mirror.

broom: Holding a broom, Putting a broom somewhere, Taking a broom from somewhere, Throwing a broom somewhere, Tidying up with a broom.

light: Fixing a light, Turning on a light, Turning off a light.

cup/glass/bottle: Drinking from a cup/glass/bottle, Holding a cup/glass/bottle of something, Pouring something into a cup/glass/bottle, Putting a cup/glass/bottle somewhere, Taking a cup/glass/bottle from somewhere, Washing a cup/glass/bottle.

closet/cabinet: Closing a closet/cabinet, Opening a closet/cabinet, Tidying up a closet/cabinet.

paper/notebook: Someone is holding a paper/notebook, Putting their paper/notebook somewhere, Taking paper/notebook from somewhere, Working on paper/notebook.

dish/dishes: Holding a dish, Putting a dish/es somewhere, Taking a dish/es from somewhere, Wash a dish/dishes.

sofa/couch: Lying on a sofa/couch, Sitting on sofa/couch.

floor: Lying on the floor, Sitting on the floor, Throwing something on the floor, Tidying something on the floor.

medicine: Holding some medicine, Taking/consuming some medicine.

television: Laughing at television, Watching television.

bed: Someone is awakening in bed, Lying on a bed, Sitting in a bed.

vacuum: Fixing a vacuum, Holding a vacuum, Taking a vacuum from somewhere.

doorknob: Fixing a doorknob, Grasping onto a doorknob.

refrigerator: Closing a refrigerator, Opening a refrigerator.

misc: Someone is awakening somewhere, Someone is cooking something, Someone is dressing, Someone is laughing, Someone is running somewhere, Someone is going from standing to sitting, Someone is smiling, Someone is sneezing, Someone is standing up from somewhere, Someone is undressing, Someone is eating something, Washing some clothes, Smiling at a book, Making a sandwich, Taking a picture of something, Walking through a doorway, Putting groceries somewhere, Washing their hands, Fixing their hair