Machine Vision and Applications (2016) 27:983–995
DOI 10.1007/s00138-016-0768-4

ORIGINAL PAPER

Collecting and annotating the large continuous action dataset

Daniel Paul Barrett[1] · **Ran Xu**[2] · **Haonan Yu**[1] · **Jeffrey Mark Siskind**[1]

Received: 18 June 2015 / Revised: 18 April 2016 / Accepted: 22 April 2016 / Published online: 21 May 2016

Abstract We make available to the community a new dataset to support action recognition research. This dataset is different from prior datasets in several key ways. It is significantly larger. It contains streaming video with long segments containing multiple action occurrences that often overlap in space and/or time. All actions were filmed in the same collection of backgrounds so that background gives little clue as to action class. We had five humans to replicate the annotation of temporal extent of action occurrences labeled with their classes and measured a surprisingly low level of intercoder agreement. Baseline experiments show that recent state-of-the-art methods perform poorly on this dataset. This suggests that this will be a challenging dataset to foster advances in action recognition research. This manuscript serves to describe the novel content and characteristics of the LCA dataset, present the design decisions made when filming the dataset, document the novel methods employed to annotate the dataset, and present the results of our baseline experiments.

Keywords Action recognition · Dataset · Video

Electronic supplementary material The online version of this article (doi:10.1007/s00138-016-0768-4) contains supplementary material, which is available to authorized users.

✉ Daniel Paul Barrett
dpbarret@purdue.edu

[1] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

[2] Computer Science and Engineering, SUNY Buffalo, Buffalo, NY, USA

1 Introduction

There has been considerable research interest in action recognition in video over the past two decades [1,2,4,5,7,8,11,13–18,20,22,27–34,37,40–42,44,45,47,49,50,52–54,57–59,61,62,64,67,68,71–79,81,83–85,87]. To support such research, numerous video datasets have been gathered. Liu et al.[39] summarize the available datasets as of 2011. These include KTH (6 classes, [58]), Weizmann (10 classes, [4]), CMU Soccer (7 classes, [9]), CMU Crowded (5 classes, [27]), UCF Sports (9 classes, [53]), UR ADL (10 classes, [44]), UM Gesture (14 classes, [38]), UCF Youtube (11 classes, [41]), Hollywood-1 (8 classes, [35]), Hollywood-2 (12 classes, [43]), MultiKTH (6 classes, [70]), MSR (3 classes, [86]), and TRECVID (10 classes, [63]). These datasets contain short clips, each depicting one of a small number of classes (3–14). Several more recent datasets also contain short clips, each depicting a single action, but with a larger number of action classes: UCF50 (50 classes, [52]), HMDB51 (51 classes, [33]), and UCF101 (101 classes, [65]). The VIRAT dataset [48] has 12 classes and longer streaming video.

Here, we introduce a new dataset called the *large continuous action dataset* (LCA). This dataset contains depictions of 24 action classes. The video for this dataset was filmed and annotated as part of the DARPA Mind's Eye program. A novel characteristic of this dataset is that rather than consisting of short clips each of which depicts a single action class, this dataset contains much longer streaming video segments that each contain numerous instances of a variety of action classes that often overlap in time and may occur in different portions of the field of view. The annotation that accompanies this dataset delineates not only which actions occur but also their temporal extent.

Many of the prior datasets were culled from video downloaded from the Internet. In contrast, the LCA dataset contains video that was filmed specifically to construct the dataset. While the video was filmed with people hired to act out the specified actions according to a general script, the fact that the video contains long streaming segments tends to mitigate any artificial aspects of the video and render the action depictions to be quite natural. Moreover, the fact that all of the video was filmed in a relatively small number of distinct backgrounds makes the dataset challenging; the background gives little clue as to the action class.

A further distinguishing characteristic of the LCA dataset is the degree of ambiguity. Most prior action recognition datasets, in fact most prior datasets for all computer vision tasks, make a tacit assumption that the labeling is unambiguous, and thus, there is a 'ground truth.' We had a team of five human annotators each annotate the entire LCA dataset. This allowed us to measure the degree of intercoder agreement. Surprisingly, there is a significant level of disagreement between humans as to the temporal extent of most action instances. We believe that such inherent ambiguity is a more accurate reflection of the underlying action recognition task and hope that the multiplicity of divergent annotations will help spur novel research with this more realistic dataset.

Another distinguishing characteristic of the LCA dataset is that some action occurrences were filmed simultaneously with multiple cameras with partially overlapping fields of view. While the cameras were neither spatially calibrated nor temporally synchronized, the fact that we have multiple annotations of the temporal extent of action occurrences may support future efforts to perform temporal synchronization after the fact. Furthermore, while most of the video was filmed from ground-level cameras with horizontal view, some of the video was filmed with aerial cameras with bird's eye view. Some of this video was filmed simultaneously with ground cameras. This may support future efforts to conduct scene reconstruction.

Some datasets are provided with official tasks and evaluation metrics. We refrain from doing so for this dataset. Instead, we leave it up to the community to make use of this dataset in a creative fashion for as many different tasks as it will be suited. Nonetheless, the LCA dataset contains sufficient information for users to compare their methods precisely with the results of the baseline experiments reported here.

2 Collection

The video for this dataset was filmed by DARPA in conjunction with Mitre and several performers from the Mind's Eye program.[1] See appendix included in the electronic supplementary material for a precise explanation of the origin of the video used in the LCA dataset and the distinction between it and that used as part of the Mind's Eye program.

Table 1 Verbs used as labels in the LCA dataset

Approach*	Drop*	Give*	Replace*
Arrive	Enter*	Hold	Run
Bury*	Exchange*	Leave	Stop
Carry*	Exit*	Pass*	Take*
Chase*	Flee*	Pick up*	Turn
Dig*	Follow*	Put down*	Walk

The starred verbs were used as part of the stage directions to the actors. The remaining verbs were not used as part of the stage directions, but occurred incidentally

The LCA dataset was filmed at several different locations, all of which were either military training facilities or facilities used to film Hollywood movies. The videos were filmed in a variety of backgrounds: several simulated country roads, several simulated safe houses, and several simulated middle-eastern urban environments. This manuscript reports a systematic annotation effort for this video which comprises 190 files as delineated in Table 1 in the appendix included in the electronic supplementary material.

The LCA dataset constitutes 2,302,144 frames and a total of 12-h, 51-min, and 16-s of video. For comparison, UCF50 has 1,330,936 frames and 13.81 h, HMDB51 has 632,635 frames and 5.85 h, UCF101 has 27 h, Hollywood-2 has 20.1 h, and VIRAT has 8.5 h. Several frames from this dataset illustrating several of the backgrounds are shown in Fig. 1.

3 Annotation

The LCA dataset contains annotations for 24 verbs, as delineated in Table 1. Figure 2 contains sample frame pairs for each of the 24 verbs. Of these, 17 verbs were used as part of the stage directions given to the actors to guide the actions that they performed. The remainder were not used as part of the stage directions but occurred incidentally. Nothing, however, precluded the actors from performing actions that could be described by other verbs. Thus, the video depicts many other actions than those annotated, including but not limited to riding bicycles, pushing carts, singing, pointing guns, arguing, and kicking balls. The only restriction, in principle, to these 24 verbs is that these were the only actions that were annotated. Identifying the presence of specific verbs in the context of many such confounding actions should present additional challenges.

[1] http://www.visint.org/.

Fig. 1 Several frames from the LCA dataset illustrating several of the backgrounds in which they were filmed

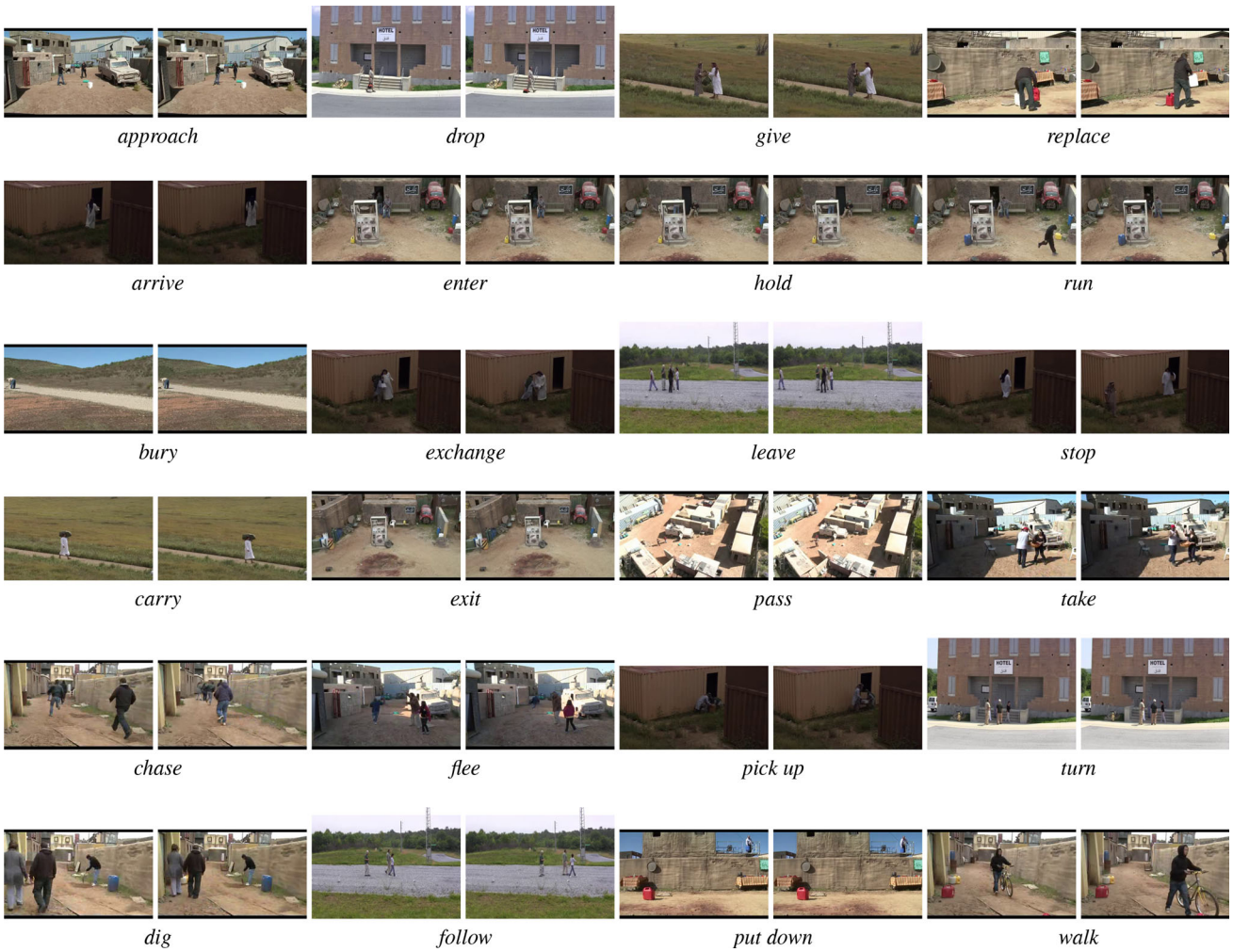

Fig. 2 Sample frame pairs from the LCA dataset illustrating the 24 action classes

We annotated all occurrences of the 24 verbs from Table 1 in the videos in Table 1 in the appendix included in the electronic supplementary material. Each such occurrence consists of a temporal interval labeled with a verb. The judgment of whether an action described by a particular verb occurred is subjective; different annotators will arrive at different judgments as to occurrence as well as the temporal extent thereof. To help guide annotators, we gave them the specification of the intended meaning of each of the 24 verbs as provided by DARPA. Annotators performed the annotation at workstations with dual monitors. One monitor displayed the annotation tool while the other monitor

displayed the documentation of intended verb meaning. The documentation of intended verb meaning is included in the LCA distribution in the electronic supplementary material.

We also asked annotators to annotate intervals where certain object classes were present in the field of view. These include *bandannas*, *bicycles*, *people*, *vehicles*, and *weapons* (the *bandannas* were worn by *people* around their head or arms). For these, a count of the number of instances of each class that were visible in the field of view was maintained. It was incremented each time a new instance became visible and decremented each time an instance became invisible. We instructed annotators that there was no need to be precise when an instance was partially visible. We further instructed annotators that *vehicles* denoted motor vehicles, not push carts, and *weapons* denoted guns, not other things like clubs or rocks that could be used as weapons.

We provided annotators with a tool that allowed them to view the videos at ordinary frame rate, stop and start the videos at will, navigate to arbitrary points in the videos, view individual frames of the videos, add, delete, and move starting and ending points of intervals, and label intervals with verbs. The tool also contained buttons to increment and decrement the counts for each of the object classes and appraised the annotator with the running counts for the object classes in each frame as the video was played or navigated.

Because of the large quantity of video to be annotated, and the fact that nothing happens during large portions of the video, we preprocessed the video to reduce the amount requiring manual annotation. We first downsampled the video to 5 fps just for the purpose of annotation; the annotation was converted back at the end to the original frame rate. Then, segments of this downsampled video where no motion occurred were removed. To do this, we computed dense optical flow on each pixel of each frame of the downsampled video. We then computed the average of the magnitude of the flow vectors in each frame and determined which frames were above a threshold. Stretches of contiguous frames that were above threshold that were separated by short stretches of contiguous frame that were below threshold were merged into single temporal segments. Then, such single temporal segments that were shorter than a specified temporal length were discarded.[2] Annotators were only given the remaining temporal segments to annotate. We performed a postprocessing step whereby the authors manually viewed all discarded frames to make sure that no actions started, ended, or spanned the omitted temporal segments. As part of this postprocessing step, the authors manually checked that none of the specified object classes entered or left the field of view during the omitted temporal segments.

We had five annotators each independently annotate the entire LCA dataset. Annotators were given initial instructions. During the annotation, annotators were encouraged to discuss their annotation judgments with the authors. The authors would then arbitrate the judgment, often specifying principles to guide the annotation. These principles were then circulated among the other annotators. The annotator instructions and principles developed through arbitration are included in the LCA distribution.

We performed a consistency check during the annotation process. Whenever an annotator completed annotation of a temporal segment, if that annotator did not annotate any intervals during that segment but other annotators did, we asked that annotator to review their annotation.

The LCA dataset contains five verb-annotation files for each of the video files in Table 1 in the appendix included in the electronic supplementary material. These have the same name as their corresponding video, but with the extension `txt`, and are located in directories named with each of the annotator codes `bmedikon`, `cbushman`, `kim861`, `nielder`, and `nzabikh`. Each line in each of these files contains a single temporal interval as a text string specifying a verb and two zero-origin nonnegative integers specifying the starting and ending frames of the interval inclusive. The LCA dataset also contains five object class annotation files for each of the video files in Table 1 in appendix included in the electronic supplementary material. These also share the filename with the corresponding video, but with the addition of the suffix `-enter-exits.txt`, and are located in the same directories named with each of the above annotator codes. Each line in each of these files contains a text string specifying an object class, an integer specifying the number of objects of that class entering or exiting the field of view (positive for entering and negative for exiting), and a single zero-origin nonnegative integer specifying the video frame.

4 Analysis

We analyzed the degree of agreement between the different annotators. To do this, we compared pairs of annotators, taking the judgments of one as 'ground truth' and computing the F1 score of the other. An interval in the annotation being scored was taken as a true positive if it overlapped some interval with the same label in the 'ground truth.' An interval in the annotation being scored was taken as a false positive if it did not overlap any interval with the same label in the 'ground truth.' An interval in the 'ground truth' was taken as a false negative if it did not overlap any interval with the same

[2] The threshold for average optical flow magnitude was 150. The threshold for ignoring short below-threshold spans when merging contiguous above-threshold frames into temporal segments was 50 frames. The threshold for ignoring short temporal segments was 15 frames.

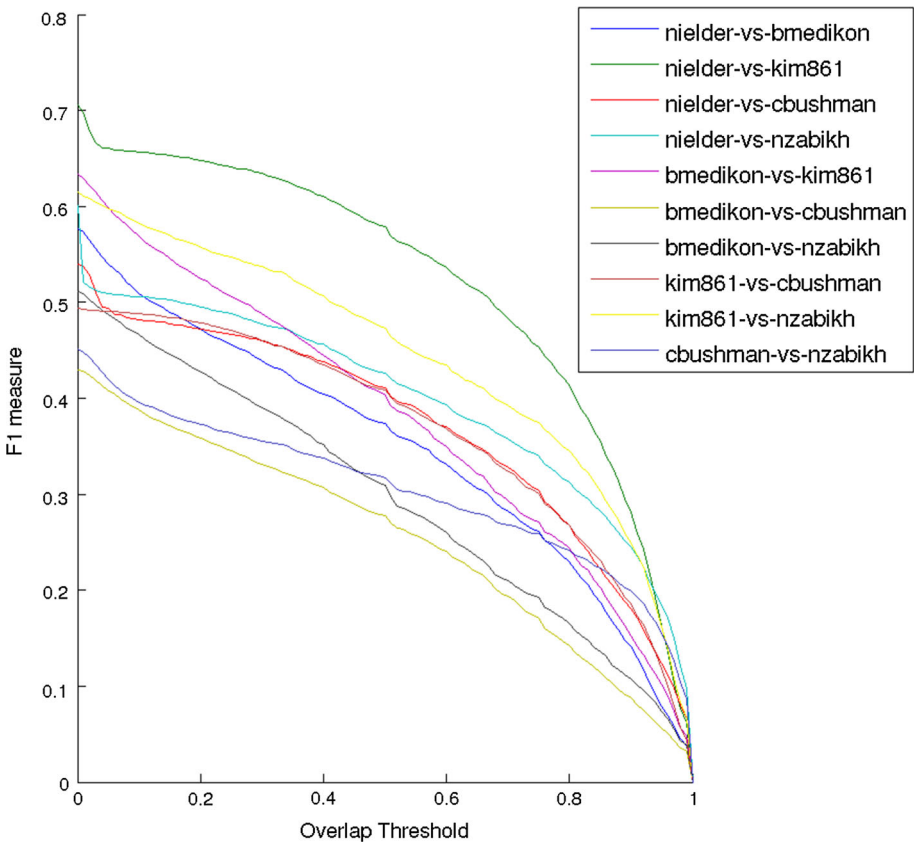

Fig. 3 Intercoder agreement on the annotations of the LCA dataset. F1 score for each pair of annotators as the overlap criterion is varied. Overlap of two intervals is measured as the length of their intersection divided by the length of their union

label in the annotation being scored. From these counts, an F1 score could be computed.

We employed the following overlap criterion. For a pair of intervals, we computed a one-dimensional variant of the 'intersection over union' criterion employed within the Pascal VOC challenge to determine overlap of two axis-aligned rectangles [10], namely the length of the intersection divided by the length of the union. We considered two intervals to overlap when the above exceeded some specified threshold. We then computed the F1 score as this threshold was varied and plotted the results for all pairs of annotators (Fig. 3).

Note that there is a surprisingly low level of agreement between annotators. Annotators rarely if ever agree on the precise temporal extent of an action as indicated by the fact that all agreement curves go to zero as the overlap threshold goes to one. At an overlap threshold of 0.5, the F1 score varies between about 0.3 and about 0.6. At an overlap threshold of 0.1, the threshold employed by VIRAT to score machines against humans, the F1 score varies between about 0.38 and about 0.67. This would put an upper bound on machine performance with this dataset using the VIRAT threshold. Even if the overlap threshold is reduced to zero, the F1 score varies between about 0.43 and about 0.7. This indicates that this dataset should be challenging for computer action recognition.

5 Baseline experiments

We performed two experiments to present and compare the performance of several state-of-the-art action recognition systems on the LCA dataset. The first experiment evaluated performance of several baseline methods on trimmed videos extracted from the LCA dataset. This task involved training and testing a classifier on a 1-out-of-24 forced-choice classification task, where each trimmed video clip nominally depicted a single action occurrence. The second experiment evaluated performance of several baseline methods on untrimmed streaming videos that comprise the entire LCA dataset. For this task, the entire LCA dataset was partitioned into five sets of videos to perform leave-one-set-out cross-validation. Models were trained on the videos in four sets and then applied to the videos in the fifth set. The task was to produce temporal intervals that delineated occurrences of the 24 action classes, each such interval labeled with the class of the action that occurred. We describe the two baseline experiments below.

The baseline experiments were performed on a collection of methods that attempt to approximate recent state-of-the-art methods for action recognition. These include running the actual released code for C2 [24], Action Bank [57], Stacked ISA [36], and VHTK [44]. We also obtained the code for Cao's method [6], Cao's reimplementation [6] of Ryoo's

method [56], and Retinotopic [3] from the authors. We also employ a number of other recent methods, including Dense Trajectories [72,73], Improved Trajectories [74], C3D [69], and the methods of Simonyan and Zisserman [60], Ng et al. [46], and Xu et al. [80].

The authors of Dense Trajectories [72,73] make their feature extractor available, but not their full pipeline [73]. Similarly, the authors of Improved Trajectories [74] also make their feature extractor available, but not their full pipeline. We reimplemented one version of a pipeline based on the original Dense Trajectories and two versions of a pipeline based on the Improved Trajectories, one that employs an SVM classifier and one that employs a neural network classifier.

This latter method, Improved Trajectories+NN, was implemented as follows. We compute Improved Trajectories for a video and then use PCA to reduce the number of dimensions of the Traj, HoG, HoF, MBHx, and MBHy features to 20, 48, 54, 48, and 48, respectively. We then train Fisher vectors [51] with a Gaussian mixture model with 32 components for each of the five features. Finally, the five Fisher vectors are concatenated to form a fixed-length feature vector of 6976 dimensions for each video. To classify these feature vectors, we construct a four-layer feed-forward neural network (4-mlp), whose structure will be described below.

We constructed a classifier based on C3D [69] by using the published pretrained models with a neural network classifier. We compute the C3D fc6 features using the model trained on the Sports-1M dataset [26]. A fixed-length 4096-dimensional feature vector is computed for every temporal interval of 16 frames in a video sample. Thus, a video sample of length T will have a sequence of $\lfloor \frac{T}{16} \rfloor$ feature vectors. All such feature vectors produced on a video sample are averaged to obtain a single feature vector for that sample. These are also classified with a 4-mlp network.

We have three additional baseline methods. The first, VGG(RGB) + PCA + VLAD, attempts to simulate Xu et al. [80]. The second, VGG(RGB) + LSTM, attempts to simulate Ng et al. [46]. Finally, both Simonyan and Zisserman [60] and Ng et al. [46] employ two data streams, one modeling image appearance features and one modeling motion through optical flow features. We simulate the latter data stream with a third baseline method, VGG(Flow) + LSTM.

The VGG(RGB) + PCA + VLAD method pools video descriptors produced by a convolutional neural network (CNN). We compute the VGG-16 fc7-relu features using the model pretrained on the ImageNet dataset [55]. A fixed-length 4096-dimensional feature vector is computed for every RGB video frame, after which the dimension is reduced to 128 with principle component analysis (PCA). We employ the vector of linearly aggregated descriptors (VLAD) method [23] with 32 K-means centers to pool the sequence of 128-dimensional feature vectors into a single 4096-dimensional feature vector per video. Again, these are also classified with a 4-mlp network.

The VGG(RGB) + LSTM method computes a sequence of VGG-16 fc7-relu feature vectors for a video, one per RGB frame. This sequence is then classified with a five-layer neural network (5-lstm) built around a Long Short-term Memory (LSTM) layer [19].

The VGG(Flow) + LSTM method is similar to the VGG (RGB) + LSTM method except that the VGG features are computed on dense optical flow fields [12], sampled at frame rate, instead of RGB frames. The same VGG model, pre-trained on the ImageNet dataset, is applied to the flow fields. The same sequence classifier, 5-lstm, is used for classifying the resulting feature vector sequence. But this classifier is retrained on the different features produced.

The 4-mlp classifiers used by Improved Trajectories + NN, C3D, and VGG(RGB) + PCA + VLAD employ the same structure. An α-dimensional input feature vector is processed by an input layer with α nodes, a first hidden layer with $\frac{\alpha}{2}$ nodes, and a second hidden layer with $\frac{\alpha}{4}$ nodes to produce an output layer with β nodes, where β is the number of classes. Similarly, the 5-lstm classifiers used by VGG(RGB) + LSTM and VGG(Flow) + LSTM also employ the same structure. An α-dimensional input feature vector is processed by an input layer with α nodes, a first hidden layer with $\frac{\alpha}{16}$ nodes, an LSTM layer with $\frac{\alpha}{16}$ nodes, and a second hidden layer with 256 nodes, to produce an output layer with β nodes, where β is the number of classes. The last instance of the output sequence of the LSTM layer is fed into the second hidden layer. All other layers are fully connected linear units. The 4-mlp and 5-lstm networks both employ a dropout layer [66] with a drop rate 0.3 before the input layer and a softmax layer after the output layer to compute the class probability. All networks employ hyperbolic tangent (tanh) as the activation function. Distinct instances of the associated network topology are trained for the different methods using stochastic gradient descent (SGD) with batch size between 10 and 20, and a learning rate between 10^{-3} and 10^{-2}.

5.1 Baseline experiment on trimmed videos

The dataset of trimmed videos was constructed from the full LCA dataset as follows. First, we took the human-annotated action intervals produced by one of the annotators, `cbushman`. This annotator was chosen to maximize the number of available action intervals. Next, a maximum of 100 intervals were selected for each action class. For those action classes for which more than 100 intervals were annotated, a random subset of 100 intervals was selected. For those action classes with 100 or fewer annotated intervals, all annotated intervals were used. A 2-s clip was extracted from the original videos centered in time on the middle of each selected anno-

Table 2 Comparison of accuracy for state-of-the-art action recognition systems on a subset of the LCA dataset with trimmed videos

Method	Accuracy (%)
Improved Trajectories + NN	18.148
VGG(Flow) + LSTM	16.666
Action Bank [57]	16.666
Improved trajectories + SVM [74]	15.556
Retinotopic [3]	14.444
Dense Trajectories [72,73]	14.074
VGG(RGB) + PCA + VLAD	10.370
C3D [69]	9.629
C2 [24]	9.259
VGG(RGB) + LSTM	8.888
Cao [6]	7.592
Cao's [6] implementation of Ryoo [56]	6.666
Stacked ISA [36]	6.666
VHTK [44]	6.296
Blind baseline (30/540)	5.555

tation interval. These clips were temporally downsampled to 20 fps and spatially downsampled to a width of 320 pixels, maintaining the aspect ratio. This process resulted in a total of 1858 clips used for the baseline experiment on trimmed videos.

The class label of each clip was considered to be the action class corresponding to the human-annotated interval from which the clip was derived. The clips for each class were randomly split into a training set with 70 % of the clips and a test set with 30 % of the clips, under the constraint that sets of clips extracted from the same video should fall completely into either the training or test set. This was done to avoid having clips from the same action (e.g., two clips from the same person digging in the same location) from appearing in both the training and test sets. This resulted in a training set of 1318 training clips and 540 test clips. Each method was trained on the training set and used to produce labels on the test set. All methods were run with default or recommended parameters. These labels were compared to the intended class labels to measure the accuracy of each method. The results of this experiment are summarized in Table 2.

There are several things of note in these results. First, all the accuracies are quite low, indicating the difficulty of the LCA dataset. The highest performing method, Improved Trajectories + NN, is correct only 18.148 % of the time. The four lowest performing methods have accuracies approaching the performance of the blind baseline (5.555 %). Additionally, many newer methods do not necessarily outperform the older methods. We suspect that this difference in relative performance of newer and older methods compared to other datasets is the result of the lack of correlation between background and action class which is often present in other datasets, as well as the presence of multiple people in the field of view. That the performance is so low and that the highest scoring methods on other datasets are not necessarily the same here shows that this dataset presents new and difficult challenges not present in other datasets.

5.2 Baseline experiment on untrimmed streaming videos

For this experiment, we employed fivefold leave-one-set-of-videos-out cross-validation. For each fold, binary classifiers using each method were trained for each action class to perform a presence/absence distinction for activity of that class. These were trained with a collection of short 2-s samples of each action class. Each presence/absence classifier was trained in a discriminative fashion with all samples from the target class in the training set as positive samples and all samples from all the other classes in the training set as negative samples. The collection of short 2-s samples of each action class was constrained to have a maximum of 100 samples for each action class. These were cropped from the streaming training videos using the temporal annotations obtained from one of the annotators, `cbushman`. For those action classes for which the training set contained fewer than 100 instances, all instances were used. For those action classes for which the training set contained more than 100 instances, 100 instances were chosen randomly.

These binary presence/absence classifiers were used to produce labeled intervals for event occurrences in the test sets as follows. Each streaming test video was divided into a sequence of 2-s clips, each overlapping the previous by 1 s. The trained binary presence/absence classifiers were used to generate confidence scores for each action class on each clip by removing the final quantization. This yields a sequence of confidence scores for each action class over time on the streaming test video. This sequence of scores was smoothed with an FIR filter by convolving with a finite signal: [0.1, 0.2.0.3, 0.4, 0.5, 0.4, 0.3, 0.2, 0.1]. Nonmaximum suppression was then performed on each smoothed score sequence to generate intervals for each peak in the smoothed score sequence. Each interval was given the score corresponding to the peak that produced that interval. The temporal extent of each interval was found by progressively extending the temporal interval forward and backward around the peak in 1-s increments until a local minimum in smoothed sequence score. Finally, for each action class, the top 50 % of such intervals were selected based on their confidence score.

Each method produced a set of labeled intervals for the test set in each cross-validation fold. Since the test sets for the cross-validation folds constitute a disjoint cover of the entire dataset, these were all pooled to yield a single set of intervals produced by each method for the entire dataset. Such a set

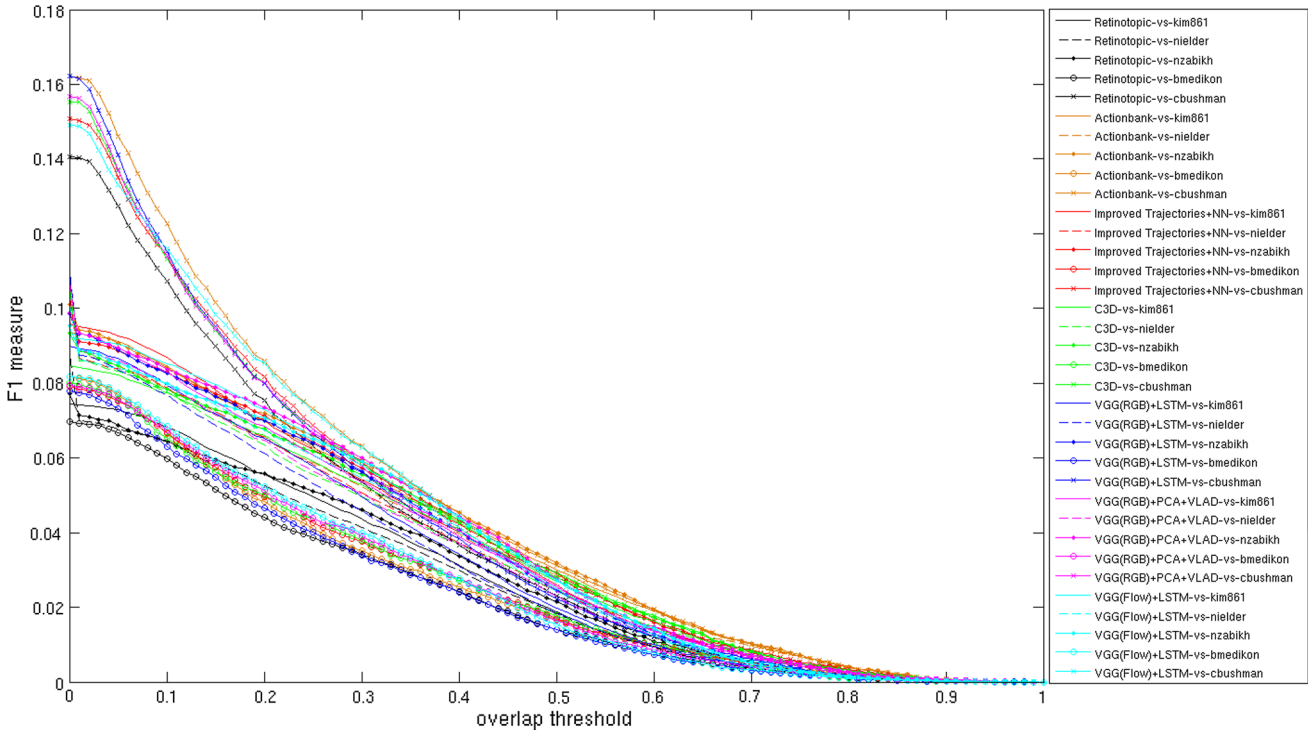

Fig. 4 Machine–human intercoder agreement on the LCA dataset. F1 score for each pair of machine methods and human annotators as the overlap criterion is varied. Overlap of two intervals is measured as the length of their intersection divided by the length of their union

of intervals produced by a method is conceptually similar to a set of intervals produced by a human annotator, since each human annotator also annotated the entire dataset. Thus, we computed machine–human intercoder agreement on sets of intervals covering the entire LCA dataset comparing each method to each human annotator using the same method for computing human–human intercoder agreement described in Sect. 4. Just as Fig. 3 characterized human–human intercoder agreement by plotting F1 score for a pair of human annotators as a function of the overlap threshold, Fig. 4 characterizes machine–human intercoder agreement by plotting F1 score between a pair of a machine method and a human annotator as a function of overlap threshold. This is done for all pairs of machine methods and human annotators. To allow comparison between machine–human and human–human intercoder agreement, Fig. 5 overlays the results of machine–human and human–human intercoder agreement. To increase legibility, machine–human and human–human intercoder agreement is shown only for pairs that include `cbushman`. Note that machine–human intercoder agreement is considerably lower than human–human intercoder agreement. The gap between machine–human and human–human intercoder agreement suggests that the LCA dataset is challenging and can serve as fodder for new action recognition research.

We similarly compared both machine–human and human–human intercoder agreement using the evaluation metric from THUMOS [25]. For each action class and every pair of interval annotations for the entire LCA dataset that included the human annotation produced by `cbushman`, we computed the average precision (AP) on the ranked set of intervals produced using five distinct overlap thresholds: 0.1, 0.2, 0.3, 0.4, and 0.5. For human annotators, we randomized the rank order as humans did not annotate intervals with confidence scores. For each overlap threshold and each pair of machine and/or human annotators, a mean average precision (mAP) was computed as an unweighted average over all action classes. Figure 6 shows the machine–human and human–human intercoder agreement using the evaluation metric from THUMOS for each overlap threshold. Again note that machine–human intercoder agreement is considerably lower than human–human intercoder agreement. The gap between machine–human and human–human intercoder agreement computed using the THUMOS evaluation criterion further supports the hypothesis that the LCA dataset is challenging and can serve as fodder for new action recognition research.

6 Related work

The THUMOS Challenge [25] also evaluates action recognition on untrimmed videos. The dataset used for the THUMOS Challenge differs from the LCA dataset in several ways. First, the THUMOS Challenge uses trimmed videos from

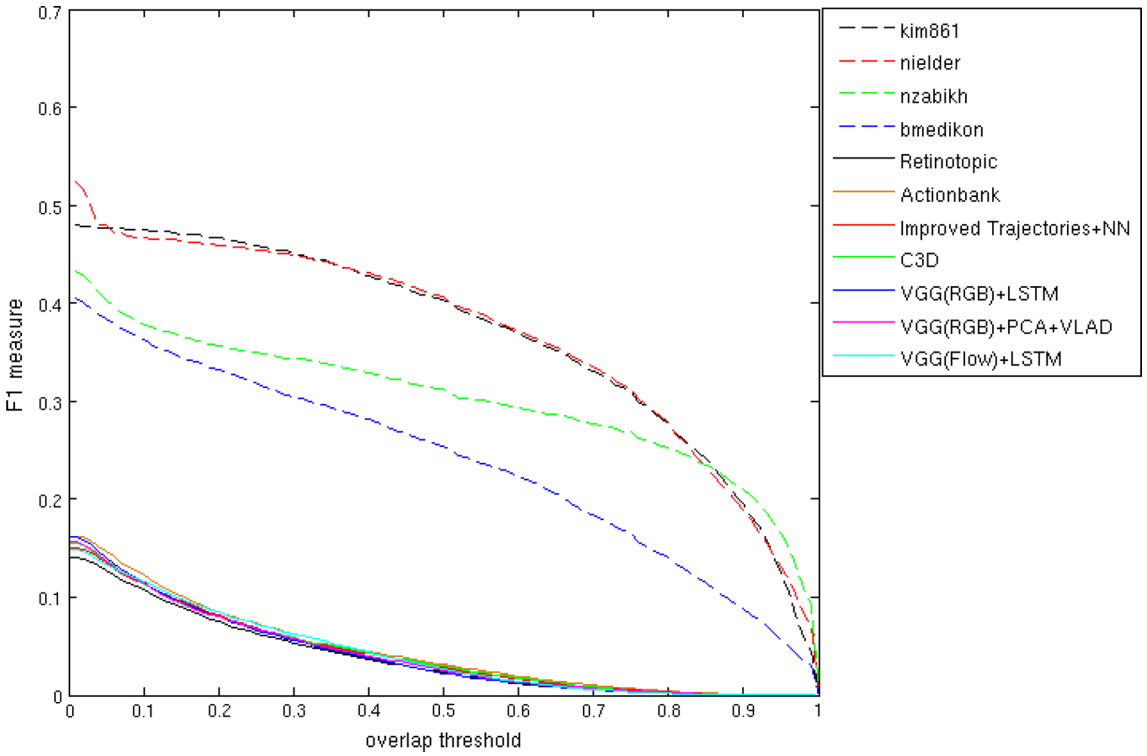

Fig. 5 Comparison between machine–human and human–human intercoder agreement on the LCA dataset, comparing against a single human annotator: `cbushman`. F1 score for each pair of annotators as the overlap criterion is varied. Overlap of two intervals is measured as the length of their intersection divided by the length of their union

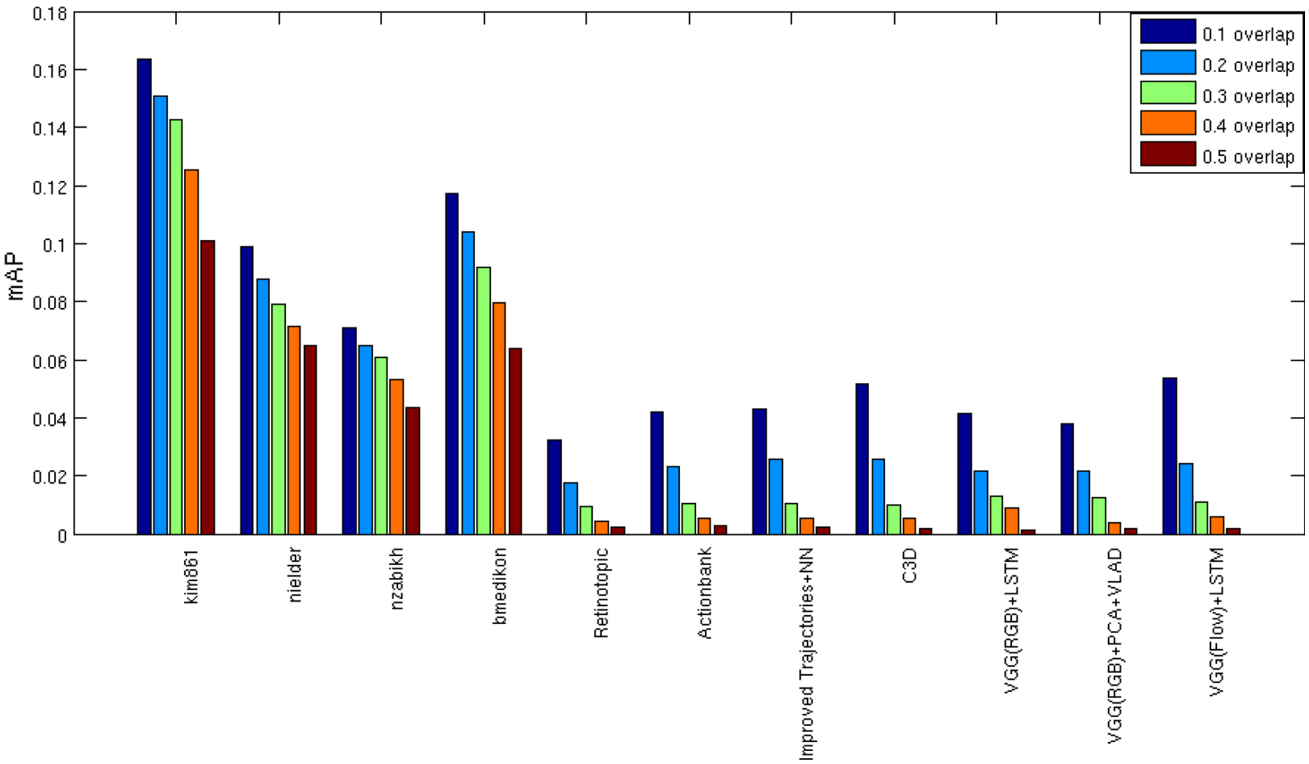

Fig. 6 Comparison between machine–human and human–human intercoder agreement on the LCA dataset, using the evaluation metric from the THUMOS Challenge [25], comparing against a single human annotator: `cbushman`

UCF101 [65] as the training set; only the validation and test sets involve untrimmed videos. The LCA dataset is partitioned into five sets of videos for training and test; each set of videos consists of untrimmed videos. Second, the validation and test sets for the THUMOS Challenge are 'untrimmed' but not 'streaming.' The videos in the LCA dataset are not just 'untrimmed' but also 'streaming.' That is, the videos in the LCA dataset are all very long, much longer than those in the THUMOS Challenge validation and test sets. Each video in the LCA dataset typically includes numerous occurrences of many different kinds of actions at different points in time and different positions in the field of view. Multiple action occurrences, of different action types, often overlap in space and/or time. Part of the intent of the LCA dataset is to evaluate the ability of methods to handle such. Third, the set of action classes used in the THUMOS Challenge are different from those used in the LCA dataset. The THUMOS Challenge uses a subset of 20 out of the 101 action classes from UCF101, all of which are sporting activities. Many of these are described by nouns rather than verbs. In contrast, the LCA dataset contains annotations for 24 action classes and five object classes. The action classes are all verbs that describe everyday activities. The action classes of the THUMOS Challenge and the LCA dataset are disjoint and cover fundamentally different kinds of activity. Fourth, the action classes in the THUMOS Challenge are often correlated with the background, *Diving* typically happens in swimming pools and *BasketballDunk* typically happens in basketball courts. The background can help distinguish activity class. In contrast, there is little to no correlation between action class and background in the LCA dataset. This forces activity recognition to rely solely on the motion characteristics of the action being performed. Thus, the THUMOS Challenge and the LCA dataset support two different kinds of research into activity recognition: methods that utilize background as part of activity recognition and methods that do not. Fifth, the THUMOS Challenge comes with a single annotation for the validation and test sets. As such, there is no way of knowing whether or not there is human agreement as to the annotation. In contrast, the LCA dataset was annotated by five different independent annotators. As such, this models the inherent ambiguity present in many natural activities. This can serve to facilitate future computer vision research that is aware of and can model such ambiguity. Finally, the action classes in the THUMOS Challenge are themselves largely unambiguous. For example, any given action occurrence is unlikely to be both *PoleVault* and *Shotput*. In contrast, the action classes in the LCA dataset overlap semantically. For example, a given action occurrence might legitimately be both an *approach* and a *walk*. That is the nature of verbs in natural language; they describe overlapping semantic classes. Moreover, there may be natural inferential structure between such semantic overlap. This inferential structure may be bidirectional. For example, it may be the case that whenever there is *chase* there is also *flee* and vice versa. The fact that *chase* and *flee* cooccur does not imply that they have the same meaning; their meaning differs in the thematic relationship between the participants. The LCA dataset can facilitate future research that studies such [82]. This inferential structure may also be unidirectional. For example, it may be the case that whenever there is *carry* there is also *walk* or *run* but not vice versa. The LCA dataset can facilitate future research that attempts to learn the inferential structure of language from visual data [21].

7 Conclusion

We make available to the community a new dataset to support action recognition research.[3] This dataset has more hours of video than HMDB51, roughly the same amount of video as UCF50, about half as much video as UCF101 and Hollywood-2, but unlike these has streaming video and has about twice as much video and twice as many classes as VIRAT, the largest dataset of streaming video. A distinguishing characteristic of this dataset is that the video is streaming; long video segments contain many actions that start and stop at arbitrary times, often overlapping in space and/or time. A further distinguishing characteristic is that while all actions were filmed in a variety of backgrounds, every action occurs in every background so that background gives little information as to action class. The above characteristics suggest that this will be a challenging dataset. This is confirmed by the low performance of recent methods on baseline experiments which also show that those methods which perform best on other datasets do not necessarily outperform other methods on this dataset. The new difficulties posed by this dataset should spur significant advances in action recognition research.

Acknowledgments This research was sponsored, in part, by the Army Research Laboratory under Cooperative Agreement Number W911NF-10-2-0060 and by NSF Grant 1522954-IIS. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Army Research Laboratory, the NSF, or the US government. The US government is authorized to reproduce and distribute reprints for government purposes, notwithstanding any copyright notation herein.

[3] http://upplysingaoflun.ecn.purdue.edu/~qobi/lca.tgz.

References

1. Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., Michaux, A., Mussman, S., Siddharth, N., Salvi, D., Schmidt, L., Shangguan, J., Siskind, J.M., Waggoner, J., Wang, S., Wei, J., Yin, Y., Zhang, Z.: Video in sentences out. In: Uncertainty in Artificial Intelligence, pp. 102–112 (2012)
2. Barbu, A., Siddharth, N., Michaux, A., Siskind, J.M.: Simultaneous object detection, tracking, and event recognition. Adv. Cogn. Syst. **2**, 203–220 (2012)
3. Barrett, D.P., Siskind, J.M.: Action recognition by time-series of retinotopic appearance and motion features. IEEE Trans. Circuits Syst. Video Technol. (2015). doi:10.1109/TCSVT.2015.2502839
4. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: International Conference on Computer Vision vol. 2, pp. 1395–1402 (2005)
5. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: Computer Vision and Pattern Recognition, pp. 994–999 (1997)
6. Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S., Siskind, J.M., Wang, S.: Recognizing human activities from partially observed videos. In: Computer Vision and Pattern Recognition, pp. 2658–2665 (2013)
7. Das, P., Xu, C., Doell, R.F., Corso, J.J.: A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In: Computer Vision and Pattern Recognition, pp. 2634–2641 (2013)
8. Dominey, P.F., Boucher, J.D.: Learning to talk about events from narrated video in a construction grammar framework. Artif. Intell. **167**(1–2), 31–61 (2005)
9. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: International Conference on Computer Vision, pp. 726–733 (2003)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
11. Everts, I., van Gemert, J.C., Gevers, T.: Evaluation of color STIPs for human action recognition. In: Computer Vision and Pattern Recognition, pp. 2850–2857 (2013)
12. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Image analysis, pp. 363–370. Springer, Berlin (2003)
13. Fernández Tena, C., Baiget, P., Roca, X., Gonzàlez, J.: Natural language descriptions of human behavior from video sequences. In: Advances in Artificial Intelligence, pp. 279–292 (2007)
14. Gaidon, A., Harchaoui, Z., Schmid, C.: Activity representation with motion hierarchies. Int. J. Comput. Vis. **107**(3), 219–238 (2014)
15. Gopalan, R.: Joint sparsity-based representation and analysis of unconstrained activities. In: Computer Vision and Pattern Recognition, pp. 2738–2745 (2013)
16. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Mooney, R., Darrell, T., Saenko, K.: Youtube2text: recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: International Conference on Computer Vision, pp. 2712–2719 (2013)
17. Gupta, A., Davis, L.S.: Objects in action: an approach for combining action understanding and object perception. In: Computer Vision and Pattern Recognition, pp. 1–8 (2007)
18. Hanckmann, P., Schutte, K., Burghouts, G.J.: Automated textual descriptions for a wide range of video events with 48 human actions. In: European Conference on Computer Vision Workshops, pp. 372–380 (2012)
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: Combining multiple features for human action recognition. In: European Conference on Computer Vision, pp. 494–507 (2010)
21. Izadinia, H., Sadeghi, F., Divvala, S.K., Hajishirzi, H., Choi, Y., Farhadi, A.: Segment-phrase table for semantic segmentation, visual entailment and paraphrasing. In: International Conference on Computer Vision, pp. 10–18 (2015)
22. Jain, A., Gupta, A., Rodriguez, M., Davis, L.S.: Representing videos using mid-level discriminative patches. In: Computer Vision and Pattern Recognition, pp. 2571–2578 (2013)
23. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition, pp. 3304–3311 (2010)
24. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: International Conference on Computer Vision, pp. 1–8 (2007)
25. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/ (2014)
26. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
27. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: International Conference on Computer Vision, pp. 1–8 (2007)
28. Khan, M.U.G., Gotoh, Y.: Describing video contents in natural language. In: Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, pp. 27–35 (2012)
29. Khan, M.U.G., Zhang, L., Gotoh, Y.: Human focused video description. In: International Conference on Computer Vision Workshops, pp. 1480–1487 (2011)
30. Khan, M.U.G., Zhang, L., Gotoh, Y.: Towards coherent natural language description of video streams. In: International Conference on Computer Vision Workshops, pp. 664–671 (2011)
31. Kojima, A., Tamura, T., Fukunaga, K.: Natural language description of human activities from video images based on concept hierarchy of actions. Int. J. Comput. Vis. **50**(2), 171–184 (2002)
32. Krishnamoorthy, N., Malkarnenkar, G., Mooney, R.J., Saenko, K., Guadarrama, S.: Generating natural-language video descriptions using text-mined knowledge. In: Conference on Artificial Intelligence, pp. 541–547 (2013)
33. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: International Conference on Computer Vision, pp. 2556–2563 (2011)
34. Laptev, I.: On space–time interest points. Int. J. Comput. Vis. **64**(2–3), 107–123 (2005)
35. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008)
36. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: Computer Vision and Pattern Recognition, pp. 3361–3368 (2011)
37. Li, L.J., Fei-Fei, L.: What, where and who? Classifying events by scene and object recognition. In: International Conference on Computer Vision, pp. 1–8 (2007)
38. Lin, Z., Jiang, Z., Davis, L.: Recognizing actions by shape-motion prototype trees. In: International Conference on Computer Vision, pp. 444–451 (2009)
39. Liu, H., Feris, R., Sun, M.T.: Benchmarking datasets for human activity recognition. In: Moeslund, T.B., Hilton, A., Krüger, V., Sigal, L. (eds.) Visual Analysis of Humans: Looking at People, Chapter 20, pp. 411–427. Springer, Berlin (2011)

40. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: Computer Vision and Pattern Recognition, pp. 3337–3344 (2011)
41. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: Computer Vision and Pattern Recognition, pp. 1996–2003 (2009)
42. Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed representation of pose and appearance. In: Computer Vision and Pattern Recognition, pp. 3177–3184 (2011)
43. Marszałek, M., Laptev, I., Schmid, C.: Actions in context. In: Computer Vision and Pattern Recognition, pp. 2929–2936 (2009)
44. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: International Conference on Computer Vision, pp. 104–111 (2009)
45. Moore, D.J., Essa, I.A., Heyes, M.H.: Exploiting human actions and object context for recognition tasks. In: International Conference on Computer Vision, pp. 80–86 (1999)
46. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Computer Vision and Pattern Recognition, pp. 4694–4702 (2015)
47. Niebles, J.C., Chen, C.W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: European Conference on Computer Vision, pp. 392–405 (2010)
48. Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.C., Lee, J.T., Mukherjee, S., Aggarwal, J.K., Lee, H., Davis, L., Swears, E., Wang, X., Ji, Q., Reddy, K., Shah, M., Vondrick, C., Pirsiavash, H., Ramanan, D., Yuen, J., Torralba, A., Song, B., Fong, A., Roy-Chowdhury, A., Desai, M.: A large-scale benchmark dataset for event recognition in surveillance video. In: Computer Vision and Pattern Recognition, pp. 3153–3160 (2011)
49. Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with Fisher vectors on a compact feature set. In: International Conference on Computer Vision, pp. 1817–1824 (2013)
50. Oreifej, O., Shah, M.: Robust Subspace Estimation Using Low-Rank Optimization, Chapter 5. Springer, Berlin (2014)
51. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: European Conference on Computer Vision, pp. 143–156 (2010)
52. Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. Mach. Vis. Appl. **24**(5), 971–981 (2013)
53. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008)
54. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: International Conference on Computer Vision, pp. 433–440 (2013)
55. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 1–42 (2015)
56. Ryoo, M.S.: Human activity prediction: early recognition of ongoing activities from streaming videos. In: International Conference on Computer Vision, pp. 1036–1043 (2011)
57. Sadanand, S., Corso, J.J.: Action bank: a high-level representation of activity in video. In: Computer Vision and Pattern Recognition, pp. 1234–1241 (2012)
58. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: International Conference on Pattern Recognition, pp. 32–36 (2004)
59. Siddharth, N., Barbu, A., Siskind, J.M.: Seeing what you're told: sentence-guided activity recognition in video. In: Computer Vision and Pattern Recognition, pp. 732–739 (2014)
60. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
61. Siskind, J.M.: Visual event classification via force dynamics. In: Conference on Artificial Intelligence, pp. 149–155 (2000)
62. Siskind, J.M., Morris, Q.: A maximum-likelihood approach to visual event classification. In: European Conference on Computer Vision, pp. 347–360 (1996)
63. Smeaton, A., Over, P., Kraaij, W.: Evaluation campaigns and TRECVID. In: ACM International Conference on Multimedia Information Retrieval, pp. 321–330 (2006)
64. Song, Y., Morency, L.P., Davis, R.: Action recognition by hierarchical sequence summarization. In: Computer Vision and Pattern Recognition, pp. 3562–3569 (2013)
65. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. Tech. Rep. arXiv:1212.0402 (2012)
66. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
67. Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: Computer Vision and Pattern Recognition, pp. 1250–1257 (2012)
68. Tian, Y., Sukthankar, R., Shah, M.: Spatiotemporal deformable part models for action detection. In: Computer Vision and Pattern Recognition, pp. 2642–2649 (2013)
69. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: International Conference on Computer Vision, pp. 4489–4497 (2015)
70. Uemura, H., Ishikawa, S., Mikolajczyk, K.: Feature tracking and motion compensation for action recognition. In: British Machine Vision Conference, pp. 1–10 (2008)
71. Wang, C., Wang, Y., Yuille, A.L.: An approach to pose-based action recognition. In: Computer Vision and Pattern Recognition, pp. 915–922 (2013)
72. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition, pp. 3169–3176 (2011)
73. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis. **103**(1), 60–79 (2013)
74. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: International Conference on Computer Vision, pp. 3551–3558 (2013)
75. Wang, Y., Mori, G.: Human action recognition by semilatent topic models. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1762–1674 (2009)
76. Wang, Z., Kuruoglu, E.E., Yang, X., Xu, Y., Yu, S.: Event recognition with time varying hidden Markov model. In: International Conference on Accoustic and Speech Signal Processing, pp. 1761–1764 (2009)
77. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: European Conference on Computer Vision, pp. 650–663 (2008)
78. Wu, X., Xu, D., Duan, L., Luo, J.: Action recognition using context and appearance distribution features. In: Computer Vision and Pattern Recognition, pp. 489–496 (2011)
79. Xu, G., Ma, Y.F., Zhang, H., Yang, S.: Motion based event recognition using HMM. In: International Conference on Pattern Recognition, pp. 831–834 (2002)
80. Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative CNN video representation for event detection. In: Computer Vision and Pattern Recognition, pp. 1798–1807 (2015)

81. Yamoto, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In: Computer Vision and Pattern Recognition, pp. 379–385 (1992)
82. Yu, H., Siddharth, N., Barbu, A., Siskind, J.M.: A compositional framework for grounding language inference, generation, and acquisition in video. J. Artif. Intell. Res. **52**, 601–713 (2015)
83. Yu, H., Siskind, J.M.: Grounded language learning from video described with sentences. In: Annual Meeting of the Association for Computational Linguistics, pp. 53–63 (2013)
84. Yuan, C., Hu, W., Tian, G., Yang, S., Wang, H.: Multi-task sparse learning with Beta process prior for action recognition. In: Computer Vision and Pattern Recognition, pp. 423–429 (2013)
85. Yuan, C., Li, X., Hu, W., Ling, H., Maybank, S.: 3D R transform on spatio-temporal interest points for action recognition. In: Computer Vision and Pattern Recognition, pp. 724–730 (2013)
86. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: Computer Vision and Pattern Recognition, pp. 2442–2449 (2009)
87. Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z.: Action recognition with Actons. In: International Conference on Computer Vision, pp. 3559–3566 (2013)

Daniel Paul Barrett received the BSCmpE degree from Purdue University in 2011. He is currently a Ph.D. student in the School of Electrical and Computer Engineering at Purdue University. His research interests include computer vision, robotics, and artificial intelligence, particularly their intersection, where a robot perceives, learns about, and acts on the world through noisy real-world camera and sensor input.

Haonan Yu is currently a Ph.D. student in the school of Electrical and Computer Engineering at Purdue University. Before that, he received his B.S. of computer science from Peking University, China. His research interests are computer vision and natural language processing. He is the recipient of the Best Paper Award of ACL 2013.

Jeffrey Mark Siskind received the B.A. degree in computer science from the Technion, Israel Institute of Technology in 1979, the S.M. degree in computer science from MIT in 1989, and the Ph.D. degree in computer science from MIT in 1992. He did a postdoctoral fellowship at the University of Pennsylvania Institute for Research in Cognitive Science from 1992 to 1993. He was an assistant professor at the University of Toronto Department of Computer Science from 1993 to 1995, a senior lecturer at the Technion Department of Electrical Engineering in 1996, a visiting assistant professor at the University of Vermont Department of Computer Science and Electrical Engineering from 1996 to 1997, and a research scientist at NEC Research Institute, Inc. from 1997 to 2001. He joined the Purdue University School of Electrical and Computer Engineering in 2002 where he is currently an associate professor.

Ran Xu received his Ph.D. in computer science from The State University of New York at Buffalo in 2015. Before that, he received his B.S. in electrical engineering from Tianjin University and B.A. in finance from Nankai University, both in 2010. His research focuses on computer vision and natural language processing.