Determining Interacting Objects in Human-Centric Activities via Qualitative Spatio-Temporal Reasoning

Hajar Sadeghi Sokeh, Stephen Gould, and Jochen Renz

The Australian National University, Canberra, ACT 0200
{hajar.sadeghi,stephen.gould,jochen.renz}@anu.edu.au

Abstract. Understanding the activities taking place in a video is a challenging problem in Artificial Intelligence. Complex video sequences contain many activities and involve a multitude of interacting objects. Determining which objects are relevant to a particular activity is the first step in understanding the activity. Indeed many objects in the scene are irrelevant to the main activity taking place. In this work, we consider human-centric activities and look to identify which objects in the scene are involved in the activity. We take an activity-agnostic approach and rank every moving object in the scene with how likely it is to be involved in the activity. We use a comprehensive spatio-temporal representation that captures the joint movement between humans and each object. We then use supervised machine learning techniques to recognize relevant objects based on these features. Our approach is tested on the challenging Mind's Eye dataset.

1 Introduction

Human activity recognition is motivated by the increasing needs of real-world applications. Some of these applications involve recognizing the type of activity or recognizing the object(s) which a person is interacting with. The behaviour of involved objects can be defined as a certain spatial and temporal pattern involving the interactions of a single or multiple actors.

A model can be learnt from a sequence of spatio-temporal features which describes how a person is behaving or interacting with an object for different activities. Accordingly one approach to recognizing activities involves acquiring concepts of what objects mean to them based on the function they perform in activities. In these methods, it is required to initially find the object(s) involved in the activity then consider the spatial changes between objects to recognize the activity. A pre-trained object detector can be used to detect the involved objects [1]. However in many activity recognition tasks, the type of object does not help in identifying the activity. For example, for action *carry*, it is not critical to know that person is carrying a box or a ball.

Spatio-temporal reasoning aims to represent and reason about spatial aspects of the world. It has been argued in AI that a person's form of spatio-temporal

reasoning is of a qualitative rather than a quantitative nature [2], and we aim to emulate this in our work. Coarse and vague qualitative representations frequently suffice for us to deal with the problems that we want to solve. For example, to know that the meal is prepared in the kitchen rather than knowing the exact coordinates for this activity.

Most qualitative approaches to spatial and temporal reasoning are based on relations between objects, such as regions or time intervals. Learning activity models can be formulated as the representation of time series data from tracking objects in videos and then, mining these time series for patterns obeying certain constraints. These patterns can be used for analysing videos, activity recognition, anomaly detection and extracting some information about the objects of interest in the video. Sridhar et al. [3, 4] proposed the qualitative spatial relations in a graph to naturally represent interactions between objects participating in video activities.

Yao et al. [5], developed a random field model that uses a structure learning method to learn the mutual context of objects and person body parts in human-object interaction activities. Their model discovered the connectivity and spatial relationships between the objects and body parts. Different to our work, Kjellstrom et al. [6] assumed that objects and actions of interests are already categorized. Then the relations are inferred from video data and represented as pairs between action and object classes like "drink-cup". The learned relations can then be used for object and action recognition.

Clearly, recognising objects and identifying activities are related tasks, and solving one informs the other. One motivation of our work is detecting activities without recognising objects. Once the activity is detected, we could add object recognition to identify the type of the involved object. The principal assumption that we make in this work is that it is the collective behaviour and interaction of objects rather than the individual behaviour that make an activity. This is the main reason why we pay more attention in our work to analysing the interactions of the objects using spatio-temporal primitives.

Detecting the relevant objects in human-object activities, regardless of the type of object or the activity, is a very difficult problem in its own right which is the main contribution we are making through this research. Initially, the interacting objects, people and moving objects, are detected. The objects are tracked and their behaviour in relation to each person is analysed. With the classification method used in our work, we are able to differentiate between relevant and irrelevant objects to the activities in each video.

The only input to our system is a video. After detecting the bounding boxes of people and possible relevant objects in each video, we extract statistical information from the computed spatio-temporal features. These features are calculated for each person-object pair. For example, if there are two people and five possible relevant objects in a video, we consider all ten possible relations between them. Then after classification, we calculate a ranked list of objects of interest. We use human-centric videos, which involve at least one person performing an

activity. In some of the videos like *running*, *jumping* and *walking*, there is no object involved in the activity, i.e., all other objects in the scene are irrelevant.

2 Spatio-Temporal Features

Our motivation for applying qualitative spatio-temporal features for determining interacting objects in human-object activities is initially analysed before detailing our method.

The changing spatial properties of objects in video and their changing relationships with other objects are often characteristic of particular activities. It is then possible to express some rules in terms of these changes or to learn activities based on similar change patterns [7, 3].

As mentioned before, qualitative features contain enough detailed information to permit recognition and reduce the importance of noise existence in real-world applications. There are many calculi defined in the field of *Qualitative Spatial Reasoning* [8] with many applications in high level interpretation of video data [4, 3, 9]. One of them is CORE-9, proposed by Cohn et al. [10].

CORE-9 is a uniform spatial representation of moving objects that integrates the important aspects of space. This model relies purely on obtaining minimal bounding rectangles of the objects in each video frame.

In CORE-9, the relevant objects and their minimum bounding rectangles are detected and tracked in order to extract qualitative information from the video. For every pair of objects per frame, nine cores are defined as shown in Fig. 1. Then the status of each core is determined and their changes over frames are analysed. All qualitative relations between the pair objects can be inferred using these nine cores.

Topology, size, distance and direction are some of the most important spatial properties of the objects that may change over time. In qualitative reasoning, the relative change of these characteristics is taken into account. In human-object activities, the relative size of interacting objects is important, regardless of their absolute size. For example, when a person is dragging a box, the size of box or person does not make any difference in the activity, since they can be close to the camera and look bigger, or far from the camera and look smaller.

In this work, we use a rectangle representation for objects relevant to the activities, since the applied detection algorithm gives us bounding rectangles for the objects. Extracted features are based on changes of spatio-temporal relations between human and object when the activity is occurring. Some of these features have been chosen from CORE-9. Fig. 1 indicates two rectangles A and B and illustrates how their projections define the nine cores in CORE-9. In our work, these two rectangles represent the bounding rectangles around the person and relevant object.

CORE-9 takes into account topology, direction, size, and distance between objects as well as changes of those relations over time. A function called "change function" is defined in [10] which is used for comparing changes in cores and intervals. This function is defined for each variable ν as $ch(\nu) \longmapsto \{<, =, >\}$

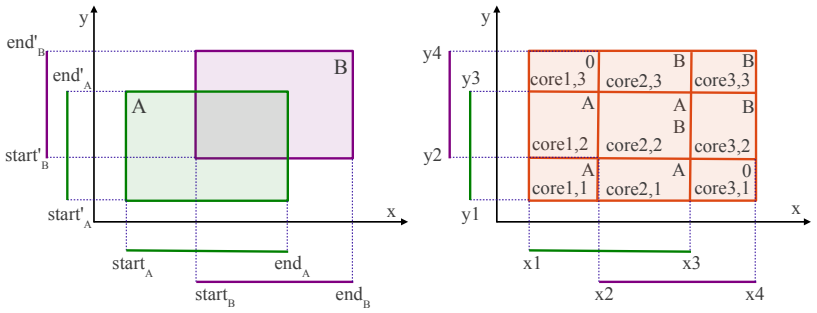


Fig. 1. Pair minimum bounding rectangles of A and B and their projections (left). Defining nine cores and six intervals using the projections in CORE-9 (right).

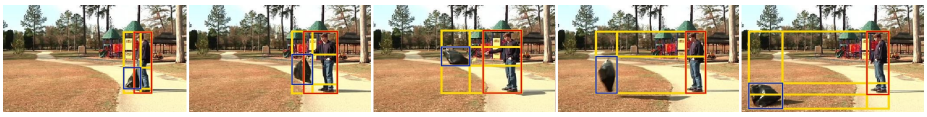


Fig. 2. (Best viewed in colour) Changing of nine cores in CORE-9 when action *throw* happens. (Nine cores of CORE-9, person and the relevant object bounding rectangles are shown by yellow, red and blue, respectively.)

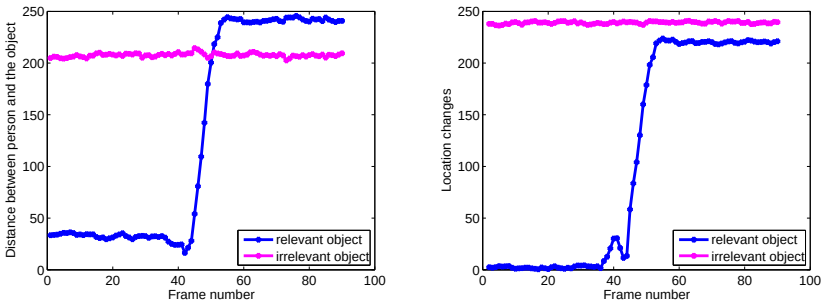
where $ch(\nu)$ is '=' if $\nu_t - \nu_{t-1} = 0$, $ch(\nu)$ is '<' if $\nu_t - \nu_{t-1} < 0$ and $ch(\nu)$ is '>' if $\nu_t - \nu_{t-1} > 0$, in which t represents a time spot in a video. The variable ν can be a core or an interval.

There are nine changes over time between sets of cores forming rectangles and six for their intervals in CORE-9. These 15 features not only give some information about the size changes for each object, but also provide some knowledge about the direction and distance changes between the bounding rectangles of two interacting objects. Fig. 2 shows how these CORE-9 features can describe the changes when a person throws a bag.

We consider distance [11] separately even though it is partially embedded in CORE-9 change features. The idea is that if a person wants to interact with an object, at least in some frames of the video they should be very close to each other and the distance between them shows this closeness.

Two other suitable features for our application are how much the location of each interacting object is changing over time. As many irrelevant moving objects like moving tree leaves, only move slightly around the same location throughout the video. Hence, these features can prune out many irrelevant objects. To remove the effect of object size on these features, we normalized the features by dividing by the object size in the image plane.

Fig. 3 shows how the change in distance and location is different for relevant and irrelevant objects. These features are calculated for the same video as Fig. 2. Regarding this figure, the distance between the person and the irrelevant object which is a small part of the tree, is not changing considerably. Instead, the



(a) Distance changes **(b)** Location changes

Fig. 3. (a) Changing distance between person and the object over frames, (b) Changing the location of the object over frames. (relevant and irrelevant objects are shown by blue and magenta respectively).

distance between the person and the bag is increasing between frames 42 and 56. There is a noteworthy difference between the location changes for relevant and irrelevant object which is illustrated in Fig. 3(b).

The prominent point here is that for the purpose of this paper, we are not interested in recognising what kind of activity is happening in the video. We track behaviour of each moving object in the scene and then label it as relevant or irrelevant. For this aim, we calculate some descriptive statistics to capture important aspects of the distribution of frame-by-frame feature changes.

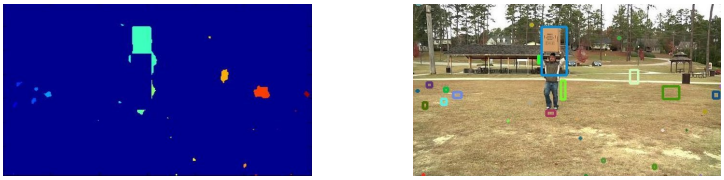
3 Detecting Human-Object Interactions

Interactions are often the main characteristic of an action. In this section, we discuss the technical steps of our method to detect these interactions.

3.1 People Detection

Detection of people is of prime importance for most activity recognition applications as many interesting activities are done by humans. In order to find the objects involved in the activities, we initially detect humans. The output of most existing people detectors is a bounding rectangle around the person which is suitable for our work.

For collecting the person detections, we use the publicly available implementation of the discriminatively trained deformable part models of Felzenszwalb et al. [12]. This algorithm has been found to outperform many others in numerous competitions. We did not have the ground truth for detections in Mind's Eye dataset, but visually the algorithm worked very well in this dataset and significant number of people were accurately detected.



(a) Optical flow **(b)** Detected blobs

Fig. 4. (Best viewed in colour) Detected moving blobs out of person bounding rectangle.

(a) activity *carry* **(b)** activity *carry* **(c)** activity *walk* **(d)** activity *jump*

Fig. 5. (Best viewed in colour) Some examples of relevant and irrelevant objects. (person: blue, relevant object: green, irrelevant object: red)

3.2 Detecting possible relevant objects

Detecting objects involved in human-object activities is a challenging problem in computer vision. In many cases, the relevant object tends to be small or only partially visible. The question here is how we can find the object of interest in each activity. In many of these interactions, a person changes the interacted object. This change can be in its shape like opening a box or in its location, for example when a person throws an object. Such changes can be detected in videos by investigating the motion of the relevant objects.

Optical flow [13] is used in this work to detect motion on all pixels of each frame except within the boundary rectangle inferred from the person detector. Fig. 4 illustrates the detected person and all possible relevant objects in a frame. There are many challenges in using optical flow for detecting the relevant objects. Tree branches moving with the wind, moving parts of human body out of its bounding rectangle, and shadow are among these challenges (see Fig. 5).

We do not consider static objects, as we can not reliably detect them with existing methods. Once static object detection works reliably, we can add these objects to our analysis without having to change our method.

If we have multiple people in the scene, we can determine relevant objects for each person separately, since our features are derived for each human-object pair.

3.3 Tracking and Data Association

After applying optical flow, there are many moving blobs in the video each of which has the possibility of being either relevant or irrelevant to the activity. The relational changes between each blob-person pair is different over time. Therefore, by analysing the spatio-temporal relational changes between each blob and person, we can differentiate between the relevant and irrelevant blobs. For the temporal analysis, each relevant object candidate should be tracked over the frames of the video.

Tracking-Learning-Detection (TLD) [14] is a real-time algorithm for tracking unknown objects in videos. Given a bounding rectangle defining the object of interest in a single frame, the algorithm automatically determines the object's bounding rectangle in other frames or indicates the invisibility of the object. TLD simultaneously tracks the object, learns its appearance and detects it whenever it appears in the video. TLD is capable of handling significant appearance changes and short-term occlusions which is very useful for real world videos such as those used in our experiments.

From the previous optical flow step, we have bounding rectangles for all moving blobs. In this step we give these bounding rectangles to the tracker as the targets to be tracked. In order to capture an object that only starts moving after the first frame, we repeat these steps for all frames of video to calculate tracks for all of possible object candidates.

In each frame we need to check if a moving blob is a new born target or an existing target which is already being tracked. Hence, we need a method to find the relationship between a detected moving blob in a frame and all detections for the previous tracks in the current frame.

A simple approach is to associate the bounding rectangles in a frame to existing targets that have the minimal Euclidean distance. In order to have more robust object association, we have applied two metrics to greatly enhance the results which are detailed as follows.

Assume B_m is a detected bounding rectangle for a moving blob in a frame. This blob can be a new born object which has just started moving in the scene. It can also represent an existing blob, if it is associated with at least one bounding rectangle in an existing track. To be considered as an existing tracked blob, the area of overlap, between B_m and the bounding rectangle of tracked object in the same frame, B_t, must exceed 80%. Bounding rectangle overlap is defined as the area of intersection divided by the area of union of the bounding rectangles. This criterion is formulated as follows.

$$overlap(B_m, B_t) = \frac{area(B_m \cap B_t)}{area(B_m \cup B_t)} > 0.8 \tag{1}$$

in which $B_m \cap B_t$ denotes the intersection of two bounding rectangles and $B_m \cup B_t$ their union.

We also extract histogram of oriented gradients [15] as a feature descriptor for both bounding rectangles, B_m and B_t. Then using a normalized square distance,

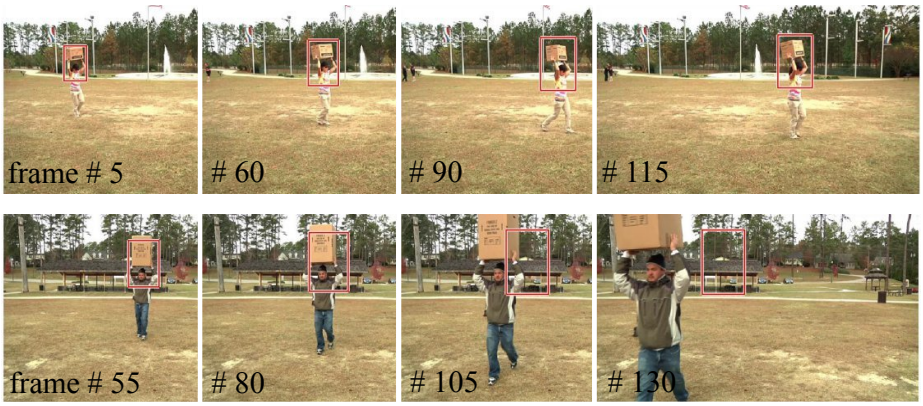


Fig. 6. Good and bad tracking results for two relevant objects in two different videos of activity *Carry*, respectively.

we quantified the difference between two bounding rectangles as another criterion with threshold of 0.05 as described in the following:

$$\frac{1}{2}\|HOG(B_m) - HOG(B_t)\|_2^2 < 0.05 \tag{2}$$

Both thresholds of 0.8 and 0.05 were chosen arbitrarily but reasonable. So, for each moving blob in a frame, we look for a bounding rectangle in one of the existing tracks which satisfies the overlap and normalized square distance measures. If there is no such B_t and no association found by the algorithm, the detected blob will be considered as a new-born object in that frame and will be tracked over the subsequent frames. Otherwise, if we find an association, we will disregard the detected moving blob as already being tracked.

The TLD tracker can continue tracking, even when there is no detection by the algorithm for a few frames, which might be due to the occlusion of the object in the scene. In such cases, we generate some linearly interpolated rectangles to represent the missed detections.

The output of this step is a track for each possible relevant object in the video. Fig. 6 illustrates some good and bad tracking results for a relevant object in two videos of activity *carry*. The second row of Fig. 6 shows how the tracking algorithm fails in some cases despite the correct initialization of the bounding box. However in the subsequent frames of this video, the object detection algorithm finds the box as a new born object again and continues tracking it.

3.4 Extracting Spatio-temporal features

As explained earlier, after detecting all possible relevant objects, we track them. We extract a feature vector of 18 different spatio-temporal features for each

human-object pair per frame, and calculate seven statistical descriptors for each spatio-temporal feature vector over frames. These descriptors are the maximum, minimum, mean, median, mode, standard deviation and variance for each feature. As a result, each spatio-temporal feature matrix is converted to a vector of $7 \times 18 = 126$ feature values. The experimental results show that these features can describe the data very well.

Our training data is highly imbalanced, as the class of relevant objects is significantly under-represented compared to the class of irrelevant objects. To overcome this problem, we over-sample positive data using Synthetic Minority Over-sampling Technique (SMOTE) [16]. In this algorithm, the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining all of the k minority class nearest neighbours.

3.5 Evaluation Algorithm

Our evaluation metric is track-based in which the metrics are computed based on each detected track and the ground truth track. We use two metrics, both on simple threshold-based correspondence. For a video, assume B_{ij} is the bounding box in the ith frame from the jth track in that video. BG_i is the bounding box of ground truth track in the ith frame in the same video. We only consider frames where both detected and ground truth objects appear. Both metrics are computed for each frame i, between a detected object track, B_{ij}, and ground truth track, BG_i. The first metric is the same overlap criteria used in data association which can be formulated as:

$$overlap(B_{ij}, BG_i) = \frac{area(B_{ij} \cap BG_i)}{area(B_{ij} \cup BG_i)} > 0.5 \tag{3}$$

In some videos, the relevant object bounding rectangle and the person bounding rectangle in one frame have too much overlap. The only clue we use to detect objects is motion in all parts of the frame except within the bounding box from the person rectangle. Then if there is too much overlap between these two rectangles, i.e. the person and the object bounding rectangle, we can only detect part of the object which is out of person bounding rectangle. In these situations, we define a second metric for the evaluation. If at least 90% of object bounding rectangle, B_{ij}, is covered by the ground truth bounding rectangle, BG_i, we consider it as a good overlapping bounding rectangle. We formulate this metric as follows:

$$\frac{area(B_{ij} \cap BG_i)}{area(B_{ij})} > 0.9 \tag{4}$$

Based on these metrics, a track is considered covered by a ground truth track if both criteria, described above, are satisfied at least for 50% of frames in which both detected and ground truth objects appear.

4 Experimental Results and Evaluation

This section outlines the train and test dataset and the obtained results using the metrics discussed in the previous section. Experiments were carried out using the challenging Mind's Eye dataset[1]. A total of 306 video sequences of 11 different activities were evaluated. The actions are: *carry*, *dig*, *fall*, *jump*, *kick*, *pickup*, *putdown*, *run*, *throw*, *turn*, *walk*. I these videos, different scenes and humans performing the activities and different objects for the same activities are used. We sampled every 5th frame in the videos and resized each frame from 720×1280 to 360×640. We also made a ground truth track corresponding to each of relevant object for each video. The number of detected tracks in all videos was 15694 which were used for training and testing the method.

After detecting and tracking the bounding rectangles for the human and all possible interacted objects per video, we extracted the qualitative spatio-temporal features for each frame of object track and the person bounding rectangle in the same frame. Next, we calculate statistical descriptors from the feature matrix, as explained in the last section, which then used for training and testing a classifier.

The key objective of this step is using a learning algorithm to build a predictive model that accurately predicts the probability of being relevant or irrelevant of previously unknown tracks. We use Support Vector Machine (SVM) in LIBSVM library [17] which is publicly available.

To our best knowledge, there is no other work on the same dataset that addresses the problem of explicit detection of relevant objects. Therefore, we defined our own baseline to compare results against. To develop a strong baseline, we have separately trained the model with each of the 18 features explained before, and distance gave the best results, 63.74% (see Table 1). This is consistent with our intuition that interacting objects exhibit characteristic patterns in how their relative distance changes over time. The next best feature was *relative speed* with performance of 55.89%. Quantitative results are shown in Table 1. As the training data is highly imbalanced, the number of *false positives* is much higher than the number of *true positives*. It gives a very low precision. We also report Macro Accuracy as the average of the true positive and true negative rates. Both numbers and percentages are included in the table from which other statistics can be derived.

To indicate the type of learning relevant objects in our work is generic and is not based on some prior knowledge about the types of activities, we performed a leave-one-activity-out cross validation experiment. We trained the classifier on 10 activities and evaluated on the 11th activity for each in turn. We measured the performance of classification for each track using the evaluation procedure detailed before. Quantitative results are presented as a confusion matrix in Table 2. The results are quite reasonable and better than the baseline as the macro accuracy shows almost 12% improvement and around 5% improvement in accuracy.

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

Table 1. The confusion matrix for the baseline algorithm.

Confusion Matrix				Precision	Macro Acc
		Predicted			
		irrelevant	relevant		
Actual	irrelevant	8394(55.33%)	6776(44.67%)	5.58%	63.74%
	relevant	146(27.86%)	378(72.14%)		

Table 2. The confusion matrix for "Leave-one-activity-out".

Confusion Matrix				Precision	Macro Acc
		Predicted			
		irrelevant	relevant		
Actual	irrelevant	11414(75.24%)	3756(24.76%)	10.49%	75.22%
	relevant	130(24.81%)	394(75.19%)		

Table 3. The confusion matrix for "10-fold cross validation".

Confusion Matrix				Precision	Macro Acc
		Predicted			
		irrelevant	relevant		
Actual	irrelevant	12293(81.03%)	2877(18.97%)	16.96%	87.08%
	relevant	36(6.87%)	488(93.13%)		

Table 4. The confusion matrix for "adding activity feature".

Confusion Matrix				Precision	Macro Acc
		Predicted			
		irrelevant	relevant		
Actual	irrelevant	13159(86.74%)	2011(13.26%)	25.46%	92.23%
	relevant	12(2.29%)	512(97.71%)		

The experimental results show that our model works quite well on unknown activities. Next, we tested our model for the case which some instances of all activities have been seen. We trained SVM with instances from all activity classes to learn how the relevant and irrelevant objects are behaving in relation to the person in each activity. We used a 10-fold cross validation on all tracks of all activities. According to the results in Table 3, almost 81% of irrelevant objects have been classified correctly as irrelevant. In this work we are interested in the number of *true positives*, i.e. the number of relevant objects which are classified correctly which in our results is more than 93%. As it is illustrated in the table, the number of irrelevant object tracks is far larger than the number of relevant objects. This imbalance in the data resulted in too many *false positives*, 2877, which is considerably more than the number of *true positives*, 488, which affects our algorithm's precision. As expected, comparing Tables 2 and 3 shows that we get more accuracy by training our model on instances from all activities.

Finally, we evaluated our model in scenarios where the activity is known. We presented a new binary indicator feature vector comprised of 11 features; the number of activities. These features were then augmented with the previous feature set. As illustrated in Table 4, using the type of activity improves the accuracy of the system significantly. Only 2.29% of the relevant data has been mistakenly classified as irrelevant and both precision and accuracy are much higher as expected. This shows that knowing the activity type provides us with more information on the relevancy of the object to the activity.

Fig. 7 shows two tracks which are correctly classified by our algorithm. The first row shows a track of an irrelevant object, a fountain in this example, which has been correctly classified. The second row also demonstrates four sequences of *true positive* tracks which were correctly classified as relevant object track.

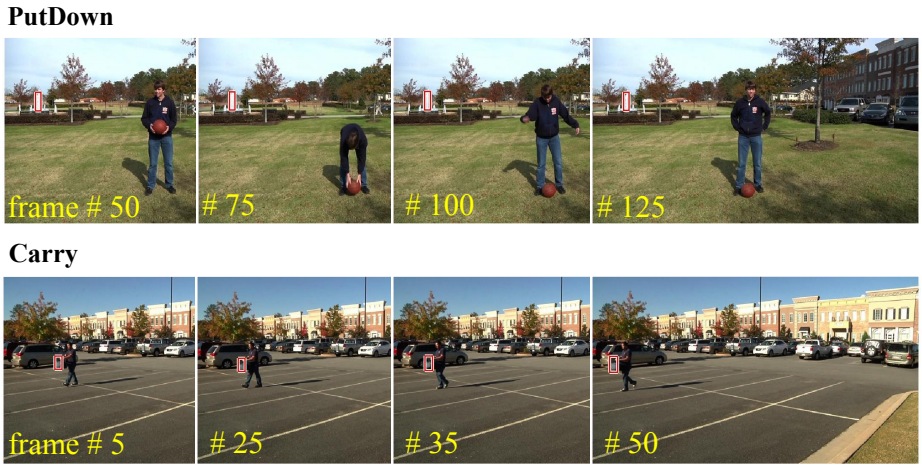


Fig. 7. Some frames of videos which our model correctly classifies. (detected relevant object is shown by red bounding rectangle). The rows are tracks of a true negative and true positive examples in the test data.

Two examples of wrong classification are illustrated in Fig. 8. The first row belongs to a *false positive* track. Based on the extracted spatio-temporal features in this work, this object which is the person's shadow, behaves like a relevant object. For these cases, we can not strongly say if they are irrelevant to the activity since they can be considered as a part of person. This problem can be addressed by applying a semantic reasoning to these tracks to classify them into the negative category.

The second row of Fig. 8 belongs to a *false negative* in which the algorithm finds this object track as an irrelevant object, whilst the evaluation algorithm finds it as a relevant object. After checking all *false negatives*, we found that more than half of them had the same problem. The problem is that the tracking algorithm failed for these objects. Consequently, after detecting the object, the

algorithm considers it as a new born target and tracks it. The new track is then classified as an irrelevant object which does not have any interaction with the person. The figure shows that in frame 35, the bag is detected as a new born target which is getting further from the person with no interaction with the person.

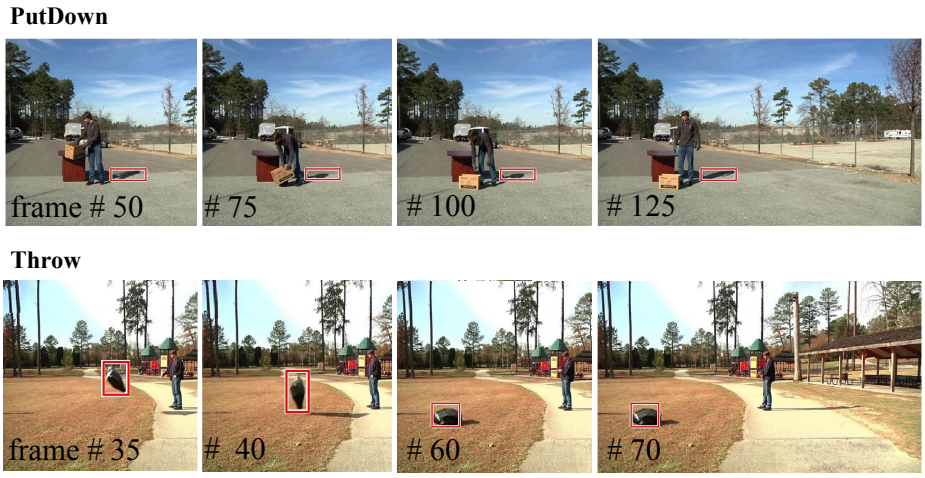


Fig. 8. Some frames of videos which our model wrongly classifies. Rows illustrates a false positive and a false negative example of test tracks, respectively.

Currently our method is implemented in Matlab. There are a number of processing steps in our pipeline. It takes 100 seconds to pre-process each frame to detect the human and extract objects. It then takes less than 1ms per human-object pair to extract the features and almost 3ms per track to be classified either as relevant or irrelevant.

5 Conclusion and Future Work

One approach to identify relevant objects to a particular activity is through qualitative spatio-temporal features. This is the main motivation of our work in this paper. We presented a framework which, given a video involving a human-centric activity, ranks every moving object with how likely it is to be involved in that activity. The extracted features are mostly about how spatial properties of objects are changing over time compared to the people. These changes mainly have a meaningful manner for the relevant objects.

We demonstrated our approach on videos involving different human activities, differentiating relevant and irrelevant objects for unknown activities. Experimental results on the real world videos demonstrate that our method works quite well without knowing the activity, but obviously can do better with the activity.

After discriminating relevant from irrelevant objects, by considering how the spatio-temporal features change between people and the relevant objects, we can recognize the activity. Therefore, one promising direction of future work is to show how this method can improve activity recognition or object recognition.

References

1. Prest, A., Ferrari, V., Schmid, C.: Explicit modeling of human-object interactions in realistic videos. Technical Report RT-0411, INRIA (2011)
2. Wolter, D., Wallgrün, J.O.: Qualitative spatial reasoning for applications: New challenges and the sparq toolbox. In: Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions. IGI Global (2010)
3. Sridhar, M., Cohn, A.G., Hogg, D.C.: Benchmarking qualitative spatial calculi for video activity analysis. In: IJCAI Workshop Benchmarks and Applications of Spatial Reasoning. (2011) 15–20
4. Sridhar, M., Cohn, A.G., Hogg, D.C.: Unsupervised learning of event classes from video. In: Association for the Advancement of Artificial Intelligence (AAAI). (2010)
5. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: Computer Vision and Pattern Recognition (CVPR). (2010) 17–24
6. Kjellström, H., Romero, J., Kragic, D.: Visual object-action recognition: Inferring object affordances from human demonstration. Computer Vision and Image Understanding **115** (2011) 81–90
7. Sokeh, H.S., Gould, S., Renz, J.: Efficient extraction and representation of spatial information from video data. In: International Joint Conferences on Artificial Intelligence (IJCAI). (2013)
8. Cohn, A.G., Renz, J. In: Qualitative Spatial Representation and Reasoning. F. van Hermelen, V. Lifschitz, B. Porter, eds., Handbook of Knowledge Representation, Elsevier (2008) 551–596
9. Sridhar, M., Cohn, A.G., Hogg, D.C.: From video to rcc8: Exploiting a distance based semantics to stabilise the interpretation of mereotopological relations. In: Conference On Spatial Information Theory (COSIT). (2011) 110–125
10. Cohn, A.G., Renz, J., Sridhar, M.: Thinking inside the box: A comprehensive spatial representation for video analysis. In: International Conference on Principles of Knowledge Representation and Reasoning (KR). (2012)
11. Hernández, D., Clementini, E., Felice, P.D.: Qualitative distances. In: Conference On Spatial Information Theory (COSIT). (1995) 45–57
12. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence (PAMI) **32** (2010) 1627–1645
13. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR. (2010) 2432–2439
14. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence (PAMI) **34** (2012) 1409–1422
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition (CVPR). (2005) 886–893
16. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16** (2002) 321–357

17. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27