Body Pose as an Indicator of Human-Object Interaction

Undergraduate Honors Thesis

Nona Sirakova
Department of Computer Science
University of Texas at Austin
nona.sirakova@utexas.edu

Supervising Professor: Dr. Kristen Grauman

Date: April 24, 2012

Submitted to The Department of Computer Science
University of Texas at Austin
In partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE IN COMPUTER SCIENCE (TURING OPTION)
April 2012

Body Pose as an Indicator of Human-Object Interaction

Undergraduate Honors Thesis

Nona Sirakova
Department of Computer Sciences
University of Texas at Austin
nona.sirakova@utexas.edu

Supervising Professor: Dr. Kristen Grauman

Date: April 24 2012

Approved by:

____
Dr. Kristen Grauman, Advisor

____
Dr. Chandrajit L Bajaj

____
Dr. Michael Walfish

Body Pose as an Indicator of Human-Object Interaction

Undergraduate Honors Thesis

Nona Sirakova
Department of Computer Sciences
University of Texas at Austin
nona.sirakova@utexas.edu

Supervising Professor: Dr. Kristen Grauman

Date: April 24 2012

Abstract

Human-object interaction recognition is important for detection of particular actions, small or partially occluded objects and video analysis. Our approach analyzes body poses and determines which ones are indicative of human-object interaction. We tested the method using videos containing different household actions. On average we obtained between 70 and 90 percent precision for the videos we tested on.

Acknowledgements:

To my adviser, Dr. Grauman, who has made this research opportunity an immensely enriching one, a fantastic learning experience, and an inspirational quest for knowledge.

To my father, who has always held my hand in my journey through science, dedicated his time to my improvement, and encouraged me through all my hardships.

To Mr. Howard Terry, who through his generous scholarship has enabled me to attend my dream university, and to be able to dedicate my time to discovering my passion for science.

Contents:

1. Introduction

Often, a human observer can deduce that a person is interacting with an object, even if the object cannot be seen well. The way a human deduces that an interaction is occurring is by observing the body pose of the subject who is performing the interaction. This observation led us to believe that there is a correlation between body poses which indicate human-object interaction, and those that indicate the absence of such an interaction.

The goal of this project is to develop an approach capable of recognizing human poses which are indicative of interaction with objects. This is achieved by developing a descriptor of the pose of a human body by first detecting a bounding box for the upper body, then detecting the face using a face detector, and finally using an upper-body pose estimator to determine the relative positions of the limbs. We then use the pose descriptor to train a learning algorithm so that it recognizes body poses that indicate the presence or absence of human-object interaction. For this study the specific instance of the object is not significant, and the algorithm does not learn how to distinguish it. Essentially, the sole important components are the body poses which indicate interaction with some object.

Whereas early work in human activity understanding focused on gesture-like actions driven entirely by the "actor" in the scene (e.g., doing jumping jacks), more recently researchers have begun to tackle the more challenging problem of understanding activities defined by how the actors interact with objects in the scene (e.g., answering a phone). To that end, several recent methods have used human pose, in combination with other features, in order to detect objects in images or video [1, 5, 6, 12, 15]. In all of these works the only interactions considered are those between humans and objects. The reason for this simplification is that human-human interaction employs a much greater variety of motions; one actor can occlude the other; and ambiguous gestures are generally involved.

In the previous works which we listed above, the object in the interaction has always been of significance. This creates the additional restriction of working with objects and poses which have been seen before.

The novelty of this thesis is that the object is irrelevant and the goal is to learn which human poses indicate interaction with an object from an arbitrary – and potentially unfamiliar – category. Some advantages of this approach are that there is no limit to the variety of the body poses; and since the algorithm does not require specific knowledge about the class of objects with which people are interacting, our approach does not suffer any setbacks if the object with which the person is interacting is partially or fully occluded, or has low resolution.

For example, in Figure 1a) and Figure 1b) the silverware and card, respectively, are small and blend in with the background. This makes them difficult to detect. In Figure 1c) the object – the blueberry in front of the boy’s eye – is too small and lacks discriminative features to be detected. In Figure 1d) the peanuts the man is sitting in form a cluttered background and blend in with the peanuts he is throwing. Thus, the peanuts he is throwing cannot be detected. In Figure 1e) and Figure 1f) the object of interaction – the toothbrush and phone – are fully or partially occluded, and therefore will be difficult to detect with an object detector. In each one of the images, it is difficult to impossible to detect the nature of the object with which the person interacts, however, the body pose of the actor is indicative of his/her action and the object he/she uses. Thus, if we can predict whether or not an interaction is likely, our system will know where and when to focus its search for an object of interest, as well as have a prior on what type of objects are most likely.

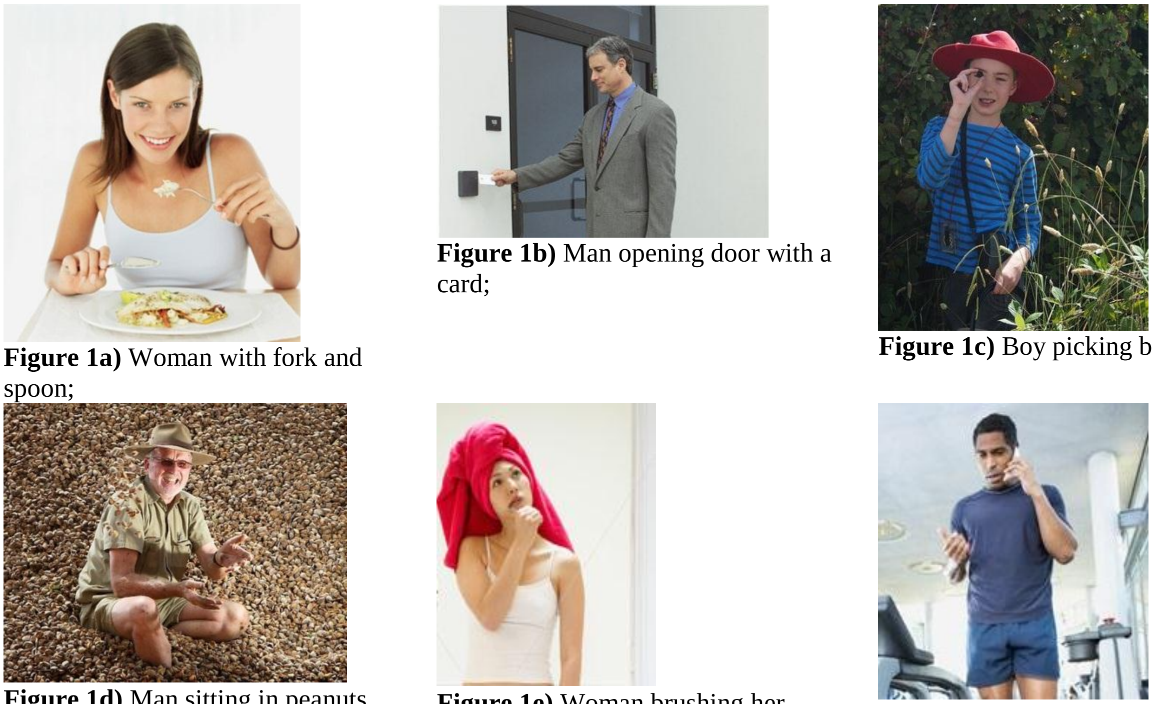


Figure 1a) Woman with fork and spoon;

Figure 1b) Man opening door with a card;

Figure 1c) Boy picking berries;

Figure 1d) Man sitting in peanuts and throwing peanuts;

Figure 1e) Woman brushing her teeth;

Figure 1f) Man talking on phone;

Figure 1a) b) c) d) e) f) Predicting whether a human-object interaction is occurring would be particularly valuable in cases where the object detection problem in isolation is too difficult, as illustrated in these images.

More generally, the ability to detect poses indicative of human-object interaction is important for activity analysis, extraction of movie frames containing action and object recognition (see Figure 1).

First, the project can be used as a stepping stone for research that aims to identify the activities which humans perform with a specific object. If given only a series of videos containing interaction with the same object, our tool will extract the frames where the human is interacting with the object. Then, different methods can be performed for action analysis, in order to learn the movements which the human performs with the object. In addition, since we maintain the relative positions of the body parts of the person in each frame, the method is able to analyze the relationship between the subject's limbs in different frames during the interaction with the object.

Second, the results of human-object interaction recognition can be used to detect where action occurs in a video. This information can be applied to compress long videos into only the parts containing action.

Finally, human-object interaction can be used to recognize objects through the body pose of a human. Often, it is the case that an object is small, partially occluded or has a small number of discriminative features. In such cases, the body pose of the human who operates the object is often indicative of the nature of the object. Such methods require a representation of the human body pose while the actor is interacting with the object. In such methods it is also necessary to be able to automatically detect frames in which interaction is likely to occur. Our method provides both a descriptor of the body pose and frames containing human-object interaction.

Therefore, there are several salient applications of the method we present in this thesis.

2. Related Work

The space of research in activity recognition and pose estimation is quite broad. We focus this review particularly on those methods that jointly estimate poses and objects for activity understanding. Approaches such as Bangpeng et al. [1] show an exception to this rule, since in this approach, the goal is to only use patches of the image, which are indicative of the action being performed. However, most approaches for action categorization consider the human pose with or without the context of the surroundings and the object the actor is interacting with.

Body pose has been used in several different scenarios, usually as a tool to obtain information for human-object interaction. For example, in [12] Prest et al. create a human centric algorithm in which they create weakly supervised leaning approach for recognizing pre learned human-object interactions. They do this by introducing a human body detector which can recognize different body parts. The algorithm then runs on a set of images of a person performing the same action and determines the special relationships between the human and the object involved with the action.

In addition, in [8] Maji et al. focus on reconstructing the unseen parts of a body in an image, and using the reconstruction, along with the visible body parts, to categorize the action a human is performing. To train their algorithm they use images where the actor's limbs, the action, and the object of interaction are all labeled. Maji et al. use those images to train their algorithm to recognize a variety of human poses. They then, introduce an unknown image, in which a part of the human limbs are cut off the edge of the image. Subsequently, Maji et al.'s trained algorithm matches the visible body parts to the most closely fitting training sample. Finally, the reconstructed body pose gets categorized as a particular action.

Thus, in both Maji and Prest's algorithms, the body pose is detected, the object of interaction is known and the goal is to characterize the action which the human is performing on the object.

Kjellstrom et al.'s approach in [6] is another work where, the body pose is considered in deciding the location of an object. In their approach Kjellstrom et al.'s extract descriptive features of the human hand and the object of interactions. The object-action correlation is modeled by using conditional random fields [6]. During classification, the change in the shape of the hands, as well as the relative movement of the hands in relation to the head are used to create a descriptor which is applied to locate the object of interaction. Next, the correlation between the descriptor and the object's shape and location are used to find an action, among the training cases, which most closely fits the data we have for the present classification example. Finally a label for the type of interaction is assigned to the novel video.

On the other hand, Peursum et al. use vivid human poses in order to classify regions of an image as specific objects [10]. This approach uses only human poses to detect objects, but the said human pose must be descriptive and unambiguous. The authors base their study on labeling patches of the image based on the objects they contain. Then they detect a rough body pose for the actor, and throughout a given video, classify the objects with which the person is interacting, based on his relationship to the patches labeled with the objects they contain.

To this end, most of the other work which aims to recognize human object interaction considers the object at hand and its relationship to the actor.

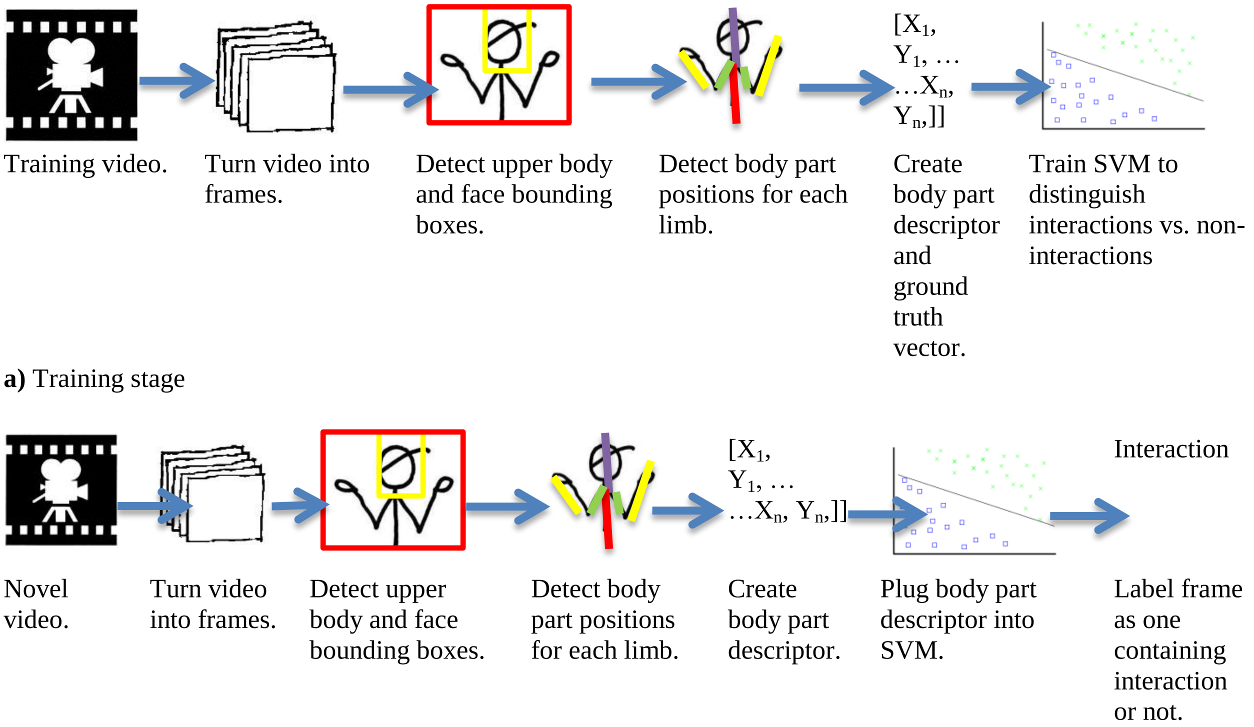
Our approach brings novelty to this field, because it is object category-independent, and only considers the human body poses which indicate object interaction in a video. Thus, unlike the case in [10] we must deal not only with very descriptive poses, but also with ambiguous ones.

3. Human-Object Interaction Detection Approach

In order to detect when human-object interaction is taking place, the main idea of our approach is to create a descriptor which works based on the location of body parts, relative to one another. We then aim to classify the descriptors into two sets – one set will indicate human-object interaction and the other will indicate the absence of such interaction. Intuitively, the set of body positions which indicate human-object interaction tend to display displacement of limbs when compared to body positions which indicate no human-object interactions. Therefore, the two sets of body poses should be able to be grouped in two clusters, sharing common characteristics. Thus, given two sets of data, where each one has its own unifying characteristics, it should be possible to teach them to a learning algorithm, so it can classify novel body poses as belonging to one of the two sets, based on common characteristics.

Briefly, our pipeline starts out by taking a video as its input and cutting it into frames. Then for each frame we find the location of the person's upper body, and use that to find the exact locations of his body parts. The body part locations are then used to form a descriptor, which is fed into the training stage of a Support Vector Machine (SVM). Once the SVM is trained, we process the video to be classified the same way we processed the videos used for training. Subsequently, we get a descriptor vector for the person in each frame of the novel video. Finally, we classify this vector using the SVM and thus, we determine if the frame contains human-object interaction.

A diagram of our pipeline's main components is shown in Figure 2.


Training video.
Turn video into frames.
Detect upper body and face bounding boxes.
Detect body part positions for each limb.
[X_1, Y_1, … …X_n, Y_n,]]
Create body part descriptor and ground truth vector.
Train SVM to distinguish interactions vs. non-interactions
Novel video.
Turn video into frames.
Detect upper body and face bounding boxes.
Detect body part positions for each limb.
[X_1, Y_1, … …X_n, Y_n,]]
Create body part descriptor.
Plug body part descriptor into SVM.
Interaction
Label frame as one containing interaction or not.

a) Training stage

b) Classification Stage

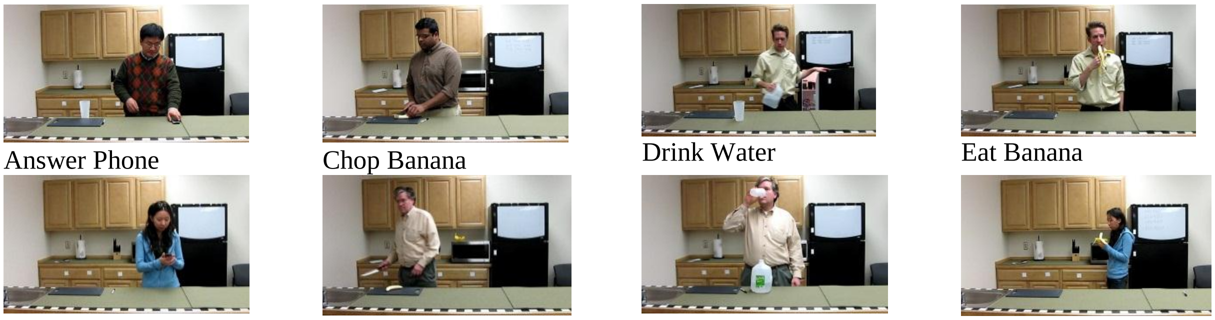
Figure 2. Overview of pipeline.

Below, we describe the test data, the input and the output of our method. We then follow with an overview of the pipeline, and finally we give a detailed description of the implementation of each step of the pipeline.

3.1 Test Data:

For training and classification data we used the University of Rochester's "Activities of Daily Living" dataset [13] which is a database of daily activity videos performed by different people and in different ways. Each video is shot through a stationary camera, and contains one person performing an assigned activity.

We used four activities – answer phone, chop banana, drink water and eat banana - each performed by five different people. In Figure 3) we show a sample of each activity. A variation of each activity was performed by each individual three times. Each video is between three and thirty seconds long. When turned into frames, each video amounts to between two hundred and six hundred frames.



Answer Phone Chop Banana Drink Water Eat Banana

Answer Phone Chop Banana Drink Water Eat Banana

Figure 3. The four actions we used, demonstrated by different actors.

3.2 Input and Output:

Once we have obtained a trained SVM from earlier stages of the pipeline, the input to our approach is a video shot through a stationary camera. This video is then processed as described in section 3.4, in order to produce a final result consisting of a labeled version of each frame of the video. A labeled frame is one which indicates for each human in the frame a prediction as to whether that human is interacting with an object or not.

3.3 Main Components of the Pipeline:

The algorithm can most generally be divided into a preparatory stage, an SVM training stage and an SVM classification stage. Hereafter, we describe the workings of each stage.

3.4 Preparatory Stage

3.4.1 **Cut the video** into individual frames by using ready library functions.

3.4.2 Detect faces

The first part of our feature descriptor relies on the position of the face in the video frame. To perform face detection in each frame we used the Viola Jones face detector [14], which briefly works as follows. Their algorithm uses a set of rectangular features which run across a potential face region of an image [14]. In the training step, the features are run on images with a known ground truth. Then a learning algorithm, similar to ADA Boost [4] is used in order to select a small set of visual features and form classifiers from those features.

Finally, a cascade is made, whose levels are classifiers. The simpler strong classifiers are towards the top of the cascade, and the more complex ones follow after them. Each level of the cascade considers only sub-windows containing features which pass through the classifiers which came higher on the cascade. The potential face regions are the ones which pass through all strong classifiers and the output we get at the end consists of the coordinates of the detected face regions. This method provides between 90% and 95% correct detection rate [14].

Thus, we chose to use this method in detecting faces because it is fast, reliable (when detecting frontal faces) and is invariant to skin color.

3.4.3 Detect the upper bodies

The next step of the pipeline is to extract the position and scale of the upper body of the actor in each video. The approach which we used to detect an upper body bounding box for a human is presented in [2]. The detector is meant for frontal view and detects human bodies based on histograms of oriented gradients. The authors detect the location of a body by detecting the face first and then narrowing down the search space for the body. In order to narrow down the search space, they separate the foreground from the background, and then search for the body, using a sliding window through the foreground of the image. In the perfect case the bounding box of the detector captures the location between the subject's head and upper arms. Because of this method's high detection rate, we used the approach provided by [3] to detect the upper body bounding boxes on our data. Some results from our experiments with the upper body detector can be seen in Figure 3. We show both a perfect detection in Figure 3a) and one with false positives in Figure 3b). Thus, as seen in Figure 3b), in a video we get many false positives and among them we have to determine the correct upper body bounding box detection.

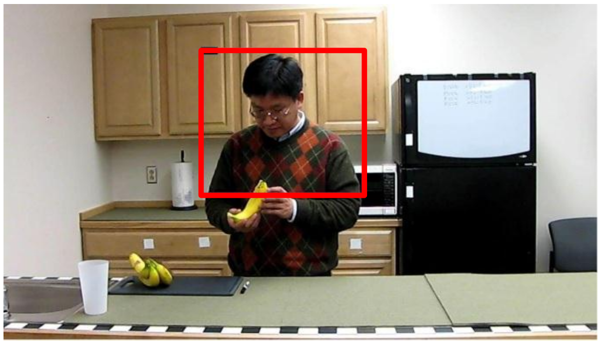


Figure 3. a) Correctly detected upper body. The bounding box for the upper body is outlined in orange.

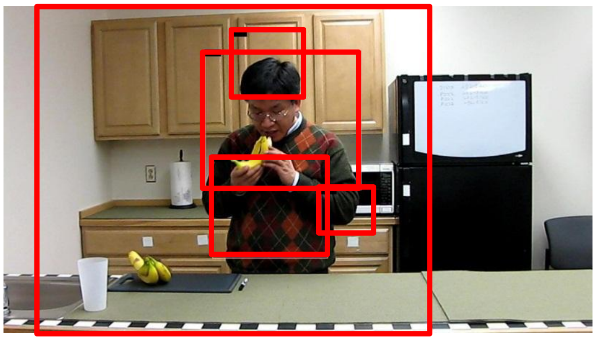


Figure 3. b) An example of false positives. There is one correct upper body bounding box and multiple false positives.

This method is helpful for finding the upper body of a human. It works well in a cluttered scene and with changing backgrounds. In addition, it provides reliable information about the scale of the person being detected. However, as mentioned above, this algorithm ends up with a high percentage of false positives. This creates a difficulty because the false positives must be eliminated in order for the results of this method to be useful.

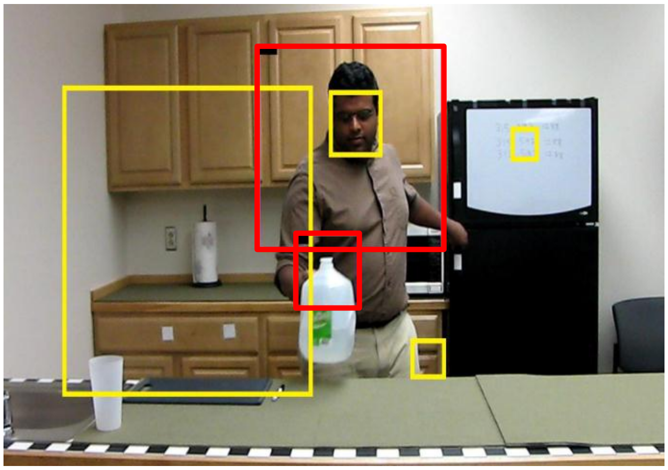


Figure 4. a) The yellow squares are the possible face detections given by the face detector. The red boxes are the possible upper body detections given by the upper body detector.

```
 x    y    w    h    box_scale    score
561  359 127  114  1.2763        3.74077372e-02
494  64   356  320   3.56239      5.15766740e-02
```

Figure 4. b) The output of the detected upper body bounding boxes. For each box, we have the following output:
x – upper x coordinate of the box.
y - upper y coordinate of the box
w - width of the box
h – height of the box,
box_scale - scale of the box
score - the probability of the box being an upper body bounding box.

Due to the high false positive rate, we must remove the false positives. The intuition is to rely more heavily on detected faces in order to prune the body detections, since face detection is a more constrained task with typically more reliable results.

Assume we are currently analyzing a given frame, and that we have found the true upper body and face bounding boxes in the previous frame. In order to find the true face bounding box and upper body bounding box in the current frame we need to go through all detections and pick the correct one.

In order to narrow down our choice of face bounding boxes in the current frame, we consider only bounding boxes which are close in location to the face bounding box we found in the previous frame. Then, for each detected upper body bounding box in the current frame we compute the area of overlap between the bounding box and the face detections which we are still considering. We, then, throw away all face bounding boxes detections which do not overlap with any upper body bounding box detections.

For the remaining set of detections, we score three features in order to determine the true upper body bounding box. The first feature we score is the percentage overlap of upper body bounding boxes with face bounding boxes. The upper body bounding box which gives the largest percent of face bounding box area inside the upper body bounding box gets scored the highest; and the upper body bounding box which contains no part of a face detection bounding box gets the lowest possible score.

The second component we score is the scale of the upper body bounding box. The closer the scale of the current upper body bounding box is to the scale of the previous frame's upper body bounding box – the higher score it gets. The way we compute score is 1/|scale of previous frame's bounding box – scale of current bounding box|.

The third factor we consider the distance of the current bounding box from the previous frame's upper body bounding box. Again, the smaller the distance between the two, the larger the score we assign for this component.

Finally, we add the three components in order to get a single numerical score for this particular upper body bounding box. Using this method, we compute the score of each upper body bounding box in the current frame. Then we pick the upper body bounding box with the highest score, to be the correct upper body bounding box for this particular frame.

The result obtained by applying this scoring method on the image shown in Figure 4 is given below, in Figure 5.

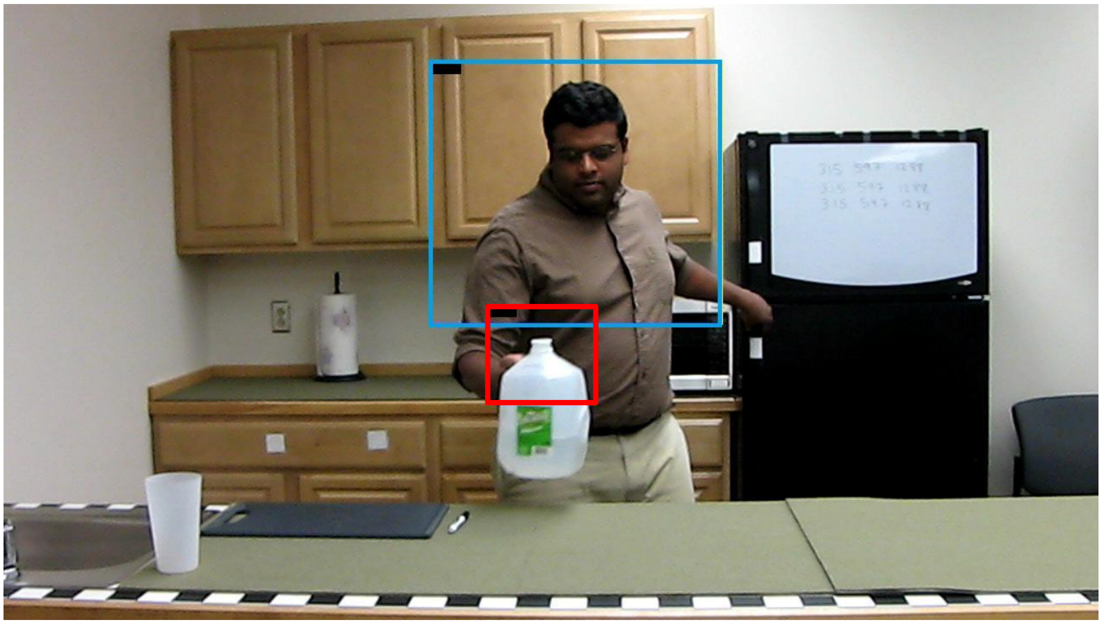


Figure 5. Shown in blue is the upper body bounding box determined by the algorithm. The box which is not selected as an upper body bounding box still appears outlined in red.

For most frames, this approach obtains the correct result as seen on Figure 5. However, in case there is a frame where none of the detected upper body bounding boxes are correct, we inevitably get the wrong result as can be seen on Figure 6. Notice, the problem arises when we do not get a good match for the upper body scale and for the distance between the upper body detection in two consecutive frames. In such cases we can resolve the issue by considering the upper body bounding box for the current frame to be the same as that of the previous frame. However, this approach fails when we have several frames in which we do not have a good upper body bounding box detection. In those cases, at the end of the K-frame long sequence of bad detections, the subject has moved far from the location where we predict to find the upper body bounding box, so even if one of the detections in the current frame is the correct bounding box for frame N+K, since our detection in frame N+K-1 was far away from the current detection, the correct current detection seems to be invalid and the method keeps not capturing the person's upper body.

As a final product of the upper body detector, for each frame it is run on, the tool returns the absolute image coordinates of all upper body bounding boxes found. In addition it outputs the probability that each detected bounding box is truly an upper body bounding box, along with the scale of each detected upper body bounding box.

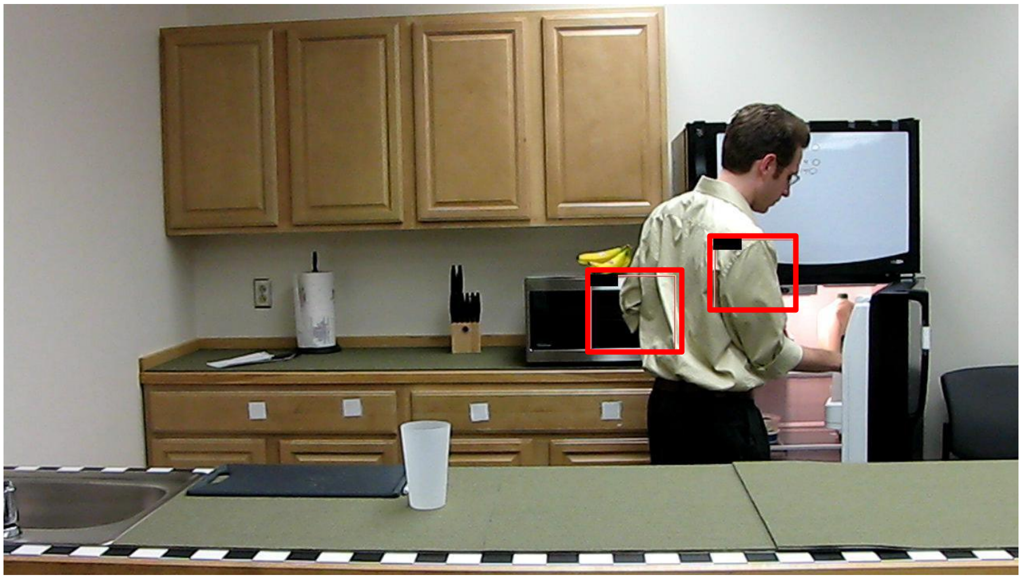


Figure 6. a) Upper body bounding boxes detected. Detected bounding boxes shown in red.

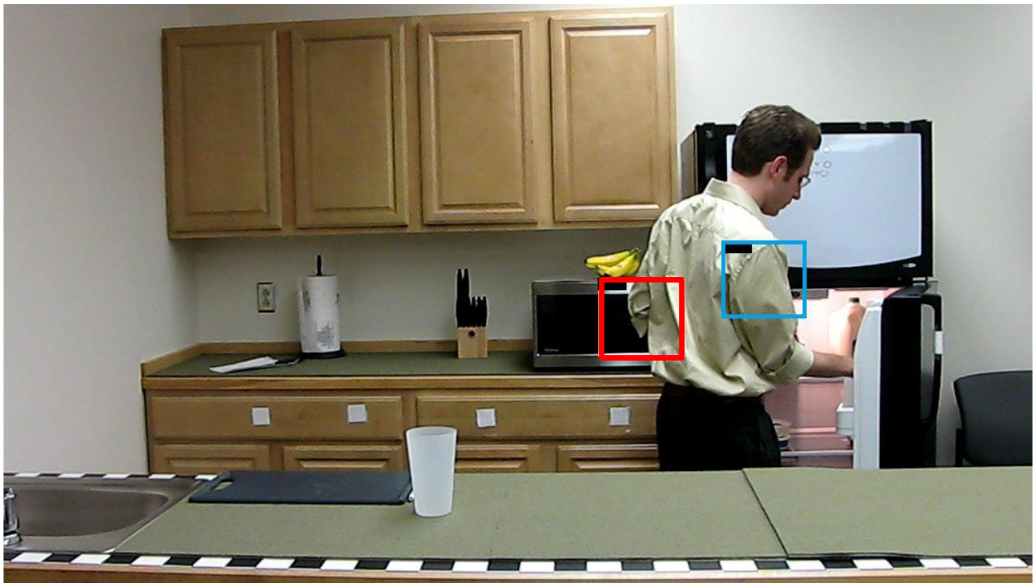


Figure 6. b) Detected upper body bounding box shown in blue.

3.4.4 **Body Parts Detector**

Having extracted the face and upper body, we next want to obtain an estimate of the 2D poses of all the body's limbs. In [3] Ferarri et al. develop a method which takes as an input an upper body bounding box, and returns the body parts of the upper body of the human in the bounding box as shown in Figure 7.

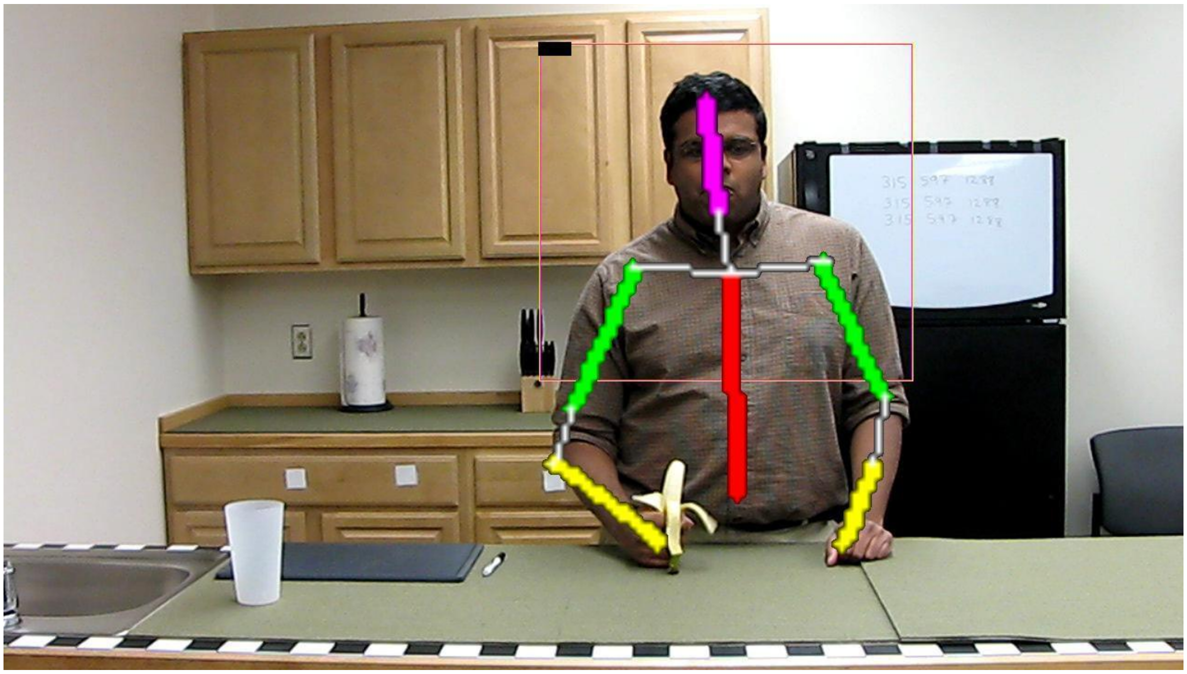


Figure 7. We use as input the upper body bounding box of the person. Then, the body parts are detected using the algorithm proposed in [3], and each limb is outlined in different color.

The method works as follows: First, the authors narrow down the search for the body parts by considering only the foreground as a possible location for the body parts. Then, given the upper body bounding box – the method reduces the search space further by using grabcut on regions detected in the upper body bounding box as weak indicators of human presence. This way, the method determines the rough locations of the head and torso. From the upper body bounding box, the authors also determine the scale of the subject. Next, Ferrari et al. limit the search space for the locations of the upper arms by considering only the parts of the foreground which are reachable by a human of the calculated scale. Finally, they detect the specific body parts by using detectors trained on those body parts.

The final output of the method are the start and end coordinates of the detected head, torso, upper left arm, upper right arm, lower left arm and lower right arm. The coordinates of the body parts are output in absolute image. In addition, to help visualize the detections, for each original frame one runs the method on, an additional frame gets created, so that that frame contains the drawn-in body parts in the locations where the method has detected them.

This approach is useful in scenes where clutter is present, there is no prior knowledge of the position of the subject to be detected, and a wide variety of poses is possible. In addition, the orientation of the subject does not have to be frontal. The authors claim that the subject can turn up to 30 degrees out of plane rotation [3]. In addition since the method does not rely on prior knowledge of skin color, the user does not have to retrain the method to tailor it to his own use.

Originally, in the body parts detector method, each detected limb is described by its starting and ending points in image coordinates. In our method, however, the actual location of the body is insignificant. We aim to be able to determine the body poses which indicate interaction no matter where the subject is in the image. Therefore we need to be able to get the limbs' location independent of their location within the frame. To solve this problem, we use as an origin the lowest point of the detected head and thus make the body parts coordinates translation invariant.

The head's coordinates are a good choice because every time we find a correct upper body bounding box, the body parts detector detects the head correctly. Thus, unlike the upper and lower arms, which do not always get detected well, the head detection gives us a reliable origin and we can form relative coordinates for each body part.

By contrast, we do not normalize the scale of the body. We do this because in the videos used for testing the relative scale of the people is about the same throughout the video. In addition, we monitor the scale of detected bodies. If we see a large change in the scale of a body from one frame to the next we take this to indicate a wrong body detection, and we do not consider this detection when training or testing the SVM.

Now, with the extraction of the relative coordinates of the body parts, we conclude the pre-processing of the videos and we are now ready to use the gathered data to train the SVM. Next, we describe the training stage of the SVM.

3.4.5 Body Pose Description Vector

We currently have a matrix containing two points (equivalent to 4 coordinates total) for each limb. In order to use the limb coordinates as vectors in an SVM, we must create a vector out of each body pose we have detected. To do so, we flatten out the array containing the relative coordinates of the body parts. Thus we get a vector, containing 24 entries (since we have 6 limbs and each limb is described by two points). The description vector will later be plugged into the SVM in our approach and algorithm.

3.5 Support Vector Machine

A Support Vector Machine (SVM) is used to assign each member of a set to one of two labels, as described in [11]. There are two stages in which the SVM achieves this goal – a training stage and a classification stage.

During the training stage, we feed into the SVM each point along with the ground truth. The ground truth indicates which set a point belongs to. Then we create a function to partition the two sets. During the classification stage, we use unknown data points, plug them into the SVM, and we observe to which set the points were assigned.

The two stages are described in more detail below.

3.5.1 Defining Ground Truth Frame Labels

As described in section 3.6.1, when training an SVM, we must plug in a set of vectors. Each vector must be labeled as belonging to one of two groups. Therefore, each frame of the videos which we will use to train our SVM must be labeled as either containing human-object interaction, or not containing human-object interaction. This information will be held in the ground truth vector which we will create as follows.

The training videos are such that each video contains a single subject who is present in the video throughout the duration of the video. Therefore, we can assume that it is enough to label each frame as one with human-object interaction, or one without human-object interaction based on the current state of the single subject in the video. Thus, if our sliced video contains N frames, the ground truth label vector contains N entries. In entry k we put a 1 if frame k of the sliced video contains human-object interaction and we put -1 otherwise.

Whether frame k contains human-object interaction is determined by a human and is therefore somewhat fuzzy. For example as a person reaches for an object, we label him as not interacting with the object until it is clear which object the person is reaching for. However, "clear" is somewhat open for interpretation. In addition, in some videos, the person is holding a knife throughout the entire video. In other words, he never reaches for the knife nor puts it down. Since we are not tracking the person's hands and their shape, in such videos, we have no way of knowing that the person is holding the knife until he starts using it. Thus, in these videos we are aware that we cannot detect the human-object interaction until the actor aims at cutting something with the knife.

An overall disadvantage of our data can be seen when constructing the ground truth vector. The disadvantage is that the videos contain a very small number of frames where the subject does not interact with any object, and a much larger set of frames where the subject does interact with an object. This trend ends up creating uneven sets of human-object interaction and non-human-object interaction and will later cause us to have to adjust our SVM.

3.5.2 Training

If the problem at hand contains n-dimensional vectors which must be classified, then a set of training points must be made first. The set of training points must possess the feature which the user wants to classify the unknown points according to. Each training point is N-dimensional and is labeled as belonging to group 1 or to group 2. The training points then get plotted into the n-dimensional SVM and a dividing hyperplane is determined between the points in group 1 and those in group 2. The SVM is now created and will be kept throughout the classification stage.

While plugging the training points it is possible to create a penalty for the hyperplane's location. In a good set of data, the hyperplane would divide the two groups in a way that all vectors of group 1 are on one side of the plane, and all vectors of group 2 are on the other. However, in some cases the data are not linearly separable and thus, there must be a compromise on the division. In this case, one can penalize the number of points which end up on the wrong side of the dividing line. This penalty approach has the effect of finding the best dividing hyperplane for a set of data which cannot be divided perfectly.

We then use the body pose descriptor vectors defined earlier, and map them in a 24 dimensional SVM. In each frame j, have one descriptor vector **j**. When we plot **j** in the SVM we label **j** as one indicating or not indicating an interaction with an object. The label we assign to **j** corresponds to the value of the j-th coordinate of the ground truth vector. Once we map the body pose descriptor vectors from all frames, the SVM space contains two sets of points. One is the set of the points which marks an interaction between the human and the object, and the other marks that there is no interaction between the human and an object.

Once the descriptors are mapped and labeled as interaction and non-interaction, we train the SVM in order to determine a separating hyper plane which best divides the two sets of points.

To deal with the imbalance in the training set, we use the standard approach of weighting label constraints for the class with fewer examples more heavily in the SVM objective. The penalty function can also depend on the amount of noise which gets introduced by frames with bad body parts detection; however we will discuss those parameter adjustments in the section 4.

In addition, since in many of the videos the subject spends a considerable amount of time facing away from the camera, in those videos much of the body parts detections are useless for training. If we keep those in the training set, often the SVM is unable to converge into two separate sets. Therefore, for the training set, we get rid of very bad detections such as the one shown in Figure 8.

Finally, we save the trained model and keep it for use when classifying a new image.

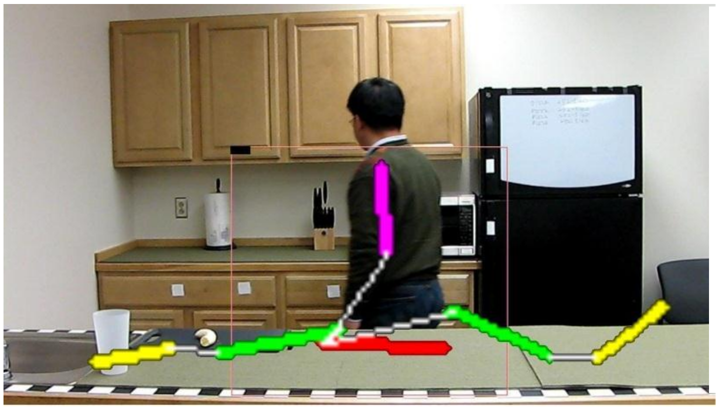


Figure 8. Outlined in red is the single upper body bounding box in the frame. Since it is the only upper body bounding box detected, it is chosen as the correct upper body bounding box. The wrong upper body bounding box gives us a detection of the body parts which is completely wrong. Such a detection would only hurt the SVM during training, therefore we exclude it from the training set.

3.5.3 Classifying a Novel Video Frame

The input to the classification stage is the SVM obtained from the training stage along with the set of n-dimensional vectors to be classified. The SVM contains group 1 and group 2, separated by a hyperplane. For each of the n-dimensional vectors to classify, in the novel test video the vector gets plugged into the SVM. Due to the vector’s coordinates and the penalty function we set during the training stage, the vector falls on one side of the hyperplane. The side it falls on determines whether the vector will be labeled as belonging to group 1 or group 2.

In our case, for the classification stage we start out by pre-processing the video in the same way we pre-processed the frames for the training stage of the training stage. First, on each frame we run the upper body detector and the face detector as we did for the preparatory stage listed above. Then, we extract the location of the upper body bounding box, and we use it to find the relative locations of the limbs and form a descriptor for the body pose, as we did in the training stage. Our next task is to predict a label according to the decision value given by the SVM classifier. Based on the set in which the new vector fell, we label it as interaction or non-interaction. Thus, in a given frame, we label people as interacting or not interacting with objects based on the way we labeled the descriptor of every person.

4. Results

We have performed 3 types of experiments which aim to demonstrate the applications of our approach in three different settings. In the first type of experiments, we train and test on videos containing the same action, performed by different people. In the second type of experiments, we train on a number of different actions, and we test on an action which was present in the training set. Finally, in the third type of experiments, we test on a set of actions, and we test on a unseen action. Note that we can expect these tests to be of increasing difficulty, because in each set we increase the variety of the training and testing data. Furthermore, the third case specifically allows us to test the category-independence of our approach, since the train and test videos will involve disjoint sets of objects.

For the experiments, we map the SVM outputs to probabilities. This allowed us to run a threshold constraint on the probability that a point belongs to a particular set. Then, for each value of the threshold we measured the precision and recall. Therefore each line on the graphs represents the results of precision versus recall for one video with varying thresholds. We will show the results of each of the three types of testing, then we will follow with a discussion on the application each type of testing is useful for and a discussion on the results.

For each experiment type, we compare our method's results for the classification of each video, to a classification performed by using a more intuitive baseline test. The workings of the baseline test are explained in section 4.1.

4.1 Baseline Test:

The baseline test consists of classifying a person as interacting or not interacting with an object based on the angle formed at the elbow by the person's forearm forms with his lower arm. We use a threshold to determine which angles indicate interaction and which ones do not. The threshold sweeps through all angles between 0 and 180 and we use the result for each value of the threshold to calculate the precision and the recall for that particular threshold. We then compare the precision/recall rates we got for this baseline test to the results we obtained from our method.

An example of the way we measure the angles is shown in Figure 9. This simple approach is a reasonable heuristic to attempt to guess whether interaction is occurring. It is important to test to verify if a more naive non-learned model is insufficient. However, we expect it to be weaker than the proposed approach, which learns from data the full space of pose descriptors that are indicative of interactions.

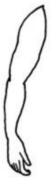


Figure 9. a) 180 degree angle between arm forearm and lower arm.

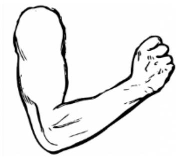


Figure 9. b) 90 degree angle between arm forearm and lower arm.

4.2 Experiments Type 1 – Train and test on same action.

4.2.1 Experiment Description

We used several videos of one activity to train the SVM. In each video, the activity is performed by a different person. This is important because each person performs the activity using different motions. For example in the activity “Answer Phone” the phone is placed at different locations and the subjects have to reach for it. In addition some people pick up the phone and put it to their left ear, whereas others put it to their right ear. Finally, some subjects put their hands on the table as they pick up the phone, while others put their free hand in their pocket.

For the classification stage of the SVM, we used a video of the same activity performed by a subject **not** seen in the training session. For the SVM training stage we removed frames which contain unusable detections, like the one shown in Figure 8. For the classification stage, each time we ran an experiment in which we removed unusable detections and we ran an experiment where all detections were used, regardless of the accuracy of the upper body detections , with the goal of understanding how much results are influenced by the errors of the component body detections.

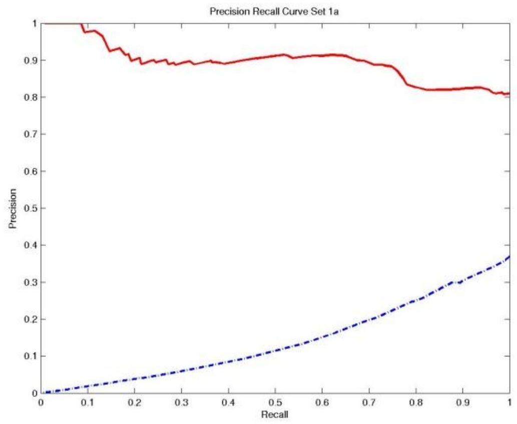


Figure 10. In red are the results of the experiments we conducted on test set 1. In blue are the results of the baseline test on the videos classified in test set 1. As evidenced by the graph our method achieves a very high precision rate, but only our method achieves a high recall rate as well.

4.2.2 Significance

In this experiment set we get high rates for both precision and recall. By comparison, the baseline test gets an almost perfect precision, but is never able to get a recall higher then 50%. This is a relatively easy scenario since we train and test on variations of the same action. Also, for the most part, the false positives and false negatives are balanced, which indicates that it is not the case that most of the points get put in the same set.

This scenario shows how the method performs in a simple setting. Yet its results could be applied to action analysis and recognition since the results indicate that when training on a particular action, the recognition of variations of that action becomes very good.

4.3 Experiments Type 2 – Train on several actions and classify a variation of the action

4.3.1 Experiment Description

To train the SVM we used several videos of different activities performed by different people. This is important because we get a greater variety of body poses which comes with the greater number of performed activities. In addition, since each activity is performed by a different subject, we eliminate the possibility that the SVM learns to distinguish one specific motion that a specific subject is making all the time. This prevents us from making an SVM which learns subject-specific information regarding the detection of human-object interaction.

For the classification stage of the SVM, we used a video of one of the training activities performed by a subject **not** used in the training session. As above, for the SVM training stage we removed frames which contain unusable detections, like the one shown in Figure 8. For the classification stage, each time we ran an experiment in which we removed unusable detections and we ran an experiment where all detections were used, regardless of the accuracy of the upper body detections.

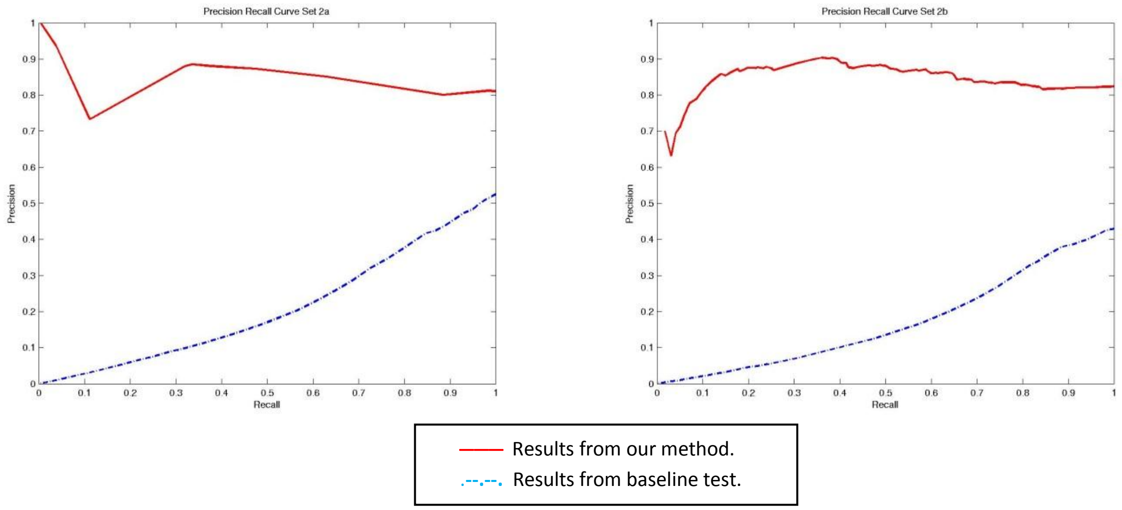


Figure 11. In these graphs, we have shown the results of our experiments of type 2 and we have compared them to the results of the same experiments performed with the baseline test. One can observe that our method's recall gets to be very high, but the precision is not as good as the one we obtain in experiment set 1 shown in Figure 10. By comparison, the baseline test gets high precision however, its recall is very unreliable.

activities and then classify a variation of one of the activities we trained on. The percent successful classification is on average 74.3% which is lower then the previous set's average of

85.4% success rate. This makes sense, because we are considering a more realistic scenario in this experiment set.

In addition, in Figure 11 we observe that the precision rate for set 2 is between 70% and 88% which is about 10% lower then the average precision obtained in experiments of type 1. However, our method still outperforms the baseline method by far.

4.4 Experiments Type 3 – Train on several actions and classify a video with an unseen action

4.4.1 Experiment Description

For this setting, we train the SVM using several videos of different activities performed by different people. Unlike the above setting, here, for the classification stage of the SVM, we used a video of an **activity** which was not seen during the training session. Once again, for the SVM training stage we removed frames which contain unusable detections, like the one shown in Figure 8. For the classification stage, we used videos which contain all detections, even the unusable ones.

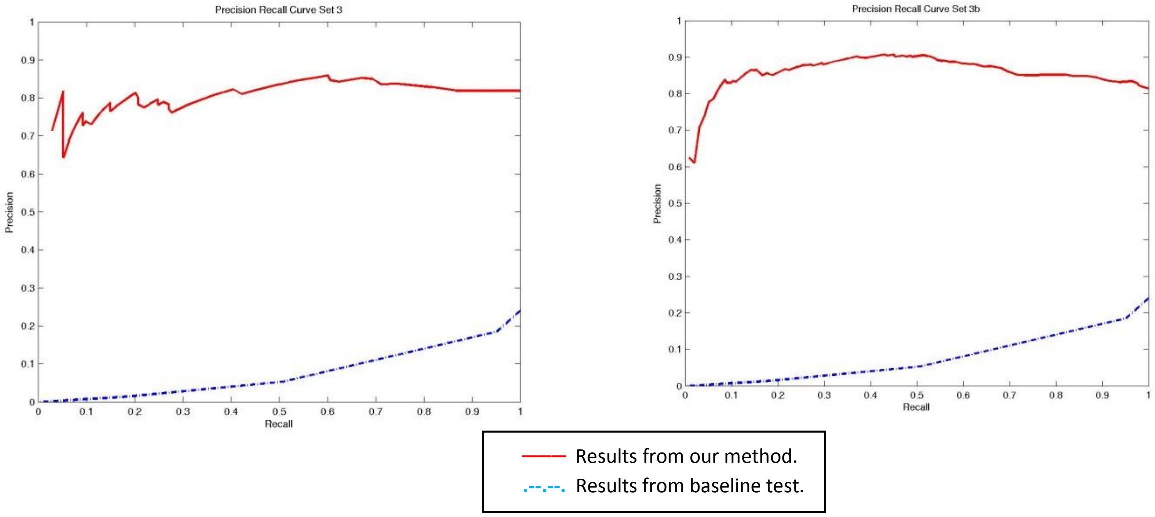


Figure 12. In these graphs, we have shown the results of our experiments of type 3 and we have compared them to the results of the same experiments performed with the baseline test. This is the most realistic and complex experiment set. This fact is mirrored in our results which show a great variation in the precision to recall ratio. However, we do get a high recall ratio and a relatively high precision ratio. By comparison, the baseline method doesn't do very well for either precision nor recall. At some threshold values, it obtains a relatively high precision, but its recall is never better then 50%.

4.4.2 **Significance**

This set of experiments represents the most realistic real world scenario because the training set does not contain the activity we classify. The percent successful classification has gone down to an average of 68.6%. Although we see a lower success rate, this experiment scenario is the most difficult scenario since the motions in the action we classify are not too similar to those in the actions we train on.

In this experiment we observe a more varied result set for our method. Precision is generally quite high, but this may be due to the higher percentage of frames containing interaction as compared to those without interaction. However, it is reassuring to see that for a while, we were able to obtain a very high recall rate. By comparison, the baseline test is also able to obtain high precision, but its recall stays below 50%.

5 Discussion of Results

Finally, we discuss the results and their meaning as a whole, focusing on the main drawbacks observed during the experiments.

For all experiments we ran, where the classification frames were filtered, the average recognition rate is 84.5%. The recognition rate of the unfiltered experiments is 55%. These results give us two types of information.

The results from the unfiltered examples show us the results one could get by running the video to be classified through our pipeline and trying to classify all frames including the ones with bad body parts detections. Since a bad body part detection can arise from wrong face detection, a bad upper body detection or a detection with a wrong location of a body part, there exist many opportunities for a wrong body part detection to arise. And because bad body parts detections are essentially noise in our data, the recognition rate of a video containing bad detections will be worse then that of the same video where the noise has been removed. Thus, the experiments with filtered classification frames show us the approximate results which we could get if we were to eliminate a large amount of the noise.

Below, we illustrate the components of the pipeline which can form noise, as well as the rationale behind the noise they form.

During the upper body bounding box detection stage we get many false positives. The upper body capture rate is at best 97% but there is a 1 to 3 ratio of false positives to true positives [3]. In Figure 3b) and Figure 6a) we saw an example of the false positives which this method detects. In examples such as Figure 3b) where we have a correct positive among the false positives, we can determine the correct upper body bounding box detection. However, in cases like Figure 6a) there is no true positive detected and therefore when choosing the upper body bounding box we are bound to make a wrong choice.

An additional component which creates faulty data for the SVM is the body parts detector. Though it is one of the best performing body parts detectors, it has some disadvantages which limit the accuracy of our results. The drawbacks of Ferrari et al.'s method for body pose detection [3] is that many of the detections it comes up with stray from the ground truth. To quantify the detection rate of the method, the authors claim that their method finds the correct estimation of the upper body of the subjects in 56% of the estimates. However they also state that they consider a "correct" detection of the body parts to be one which "lies within 50% of the length of the ground-truth segment" which allows for deviations in the detection.

To illustrate the effects of the detection inaccuracies one can observe the detection in Figure 8. where the detected body is completely different from the actual location of the person. A milder case of wrong detection can be seen in Figure 13b). In fact, throughout the experiments we noticed that often the lower arms of people get detected to be in completely different locations than they can be seen on an image. In addition, it often happens that a body part gets detected to be far shorter then it is in the real image. This usually occurs when people hold their arms in front of their torso. However, since many of the actions people perform in our videos are based on working with their hands, capturing the lower arms of the subjects is crucial.

Another drawback of this method is that since the method employs temporal dependence, it is the case that when a body part gets detected in a wrong location in one frame, it keeps getting detected in the same wrong location for many frames to come.

Another disadvantage is that as people turn more then 30 degrees, the body parts detections quickly degrade into unusable detections. Although, the researchers in [3] warn against this, in natural actions, people often end up turning from the camera.

Some examples of good and bad detections of the limbs can be seen in Figure 13. The detections were obtained by using the algorithm of [3].

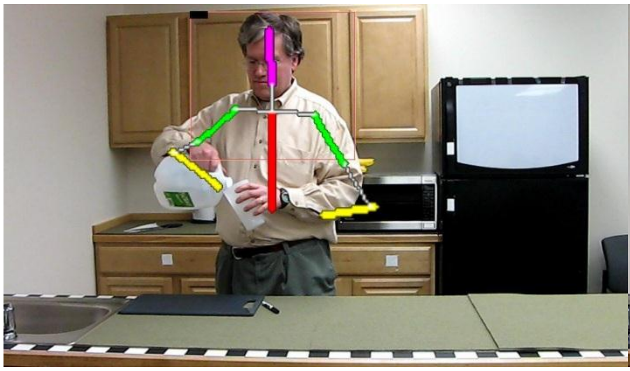


Figure 13. a) A good capturing of the upper body of the person. Notice all limbs are approximately well detected.

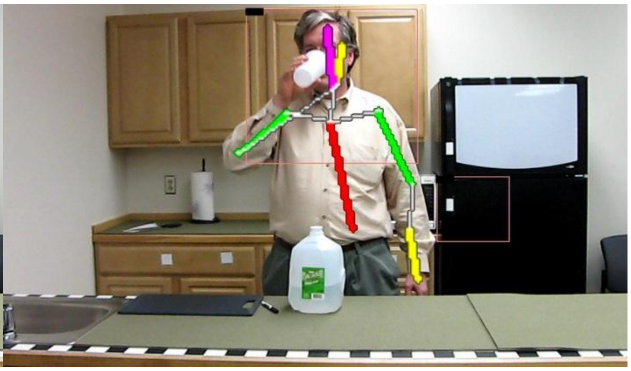


Figure 13. B) the lower right arm of the person is incorrectly detected. The detection for the lower left arm is the yellow line in front of the person's face. Notice that the confusion in detection happens on a limb which is in front of the torso/head of the person, as we mentioned above.

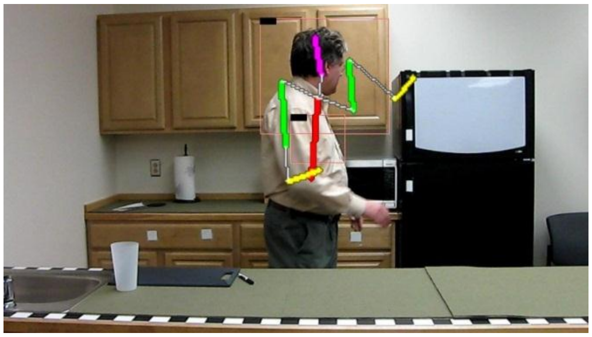


Figure 13. c) The person is turned 90 degrees away from the camera. In such cases we get a wrong detection of the upper body. However in actions such as opening the fridge and writing on the white board, the person has to turn away from the camera.

Finally, the SVM has several drawbacks: First, the results it yields are heavily dependent on the hyperparameter that penalizes the use of slack variables. Thus, in a set of truly unknown vectors which must be classified, one can obtain varying classifications. And second the SVM can only perform if in the training stage, there can be found a hyper plane which divides the vectors of group 1 and group 2. In large sets containing much noise, this is a difficult task and worsens the final results of the classification stage. To mitigate the chance of this happening, one could explore alternate feature spaces and/or kernel functions for the SVM.

Since we use only frames with good detections for the training stage of the SVM, most of the time we get good divisions between the set of training images containing object interaction and the ones not containing object interaction. Thus, when for the classification stage we pass in the frames to be classified, even in a video containing bad body parts detections, most of the good body parts detections get detected correctly. The low rate of recognition of human-object interaction in the non-filtered videos comes from the frames with bad body parts detections.

Therefore it is not surprising that the results from the videos with filtered frames, in other words – videos where the frames with bad body parts detections were discarded – gave a significantly better detection rate. These experiments were conducted in order to show the results which could be obtained if the body parts detections were less noisy then the ones we obtained.

There are two ways to increase the accuracy of the detections we obtained, and thus to get results from unfiltered videos, which are closer to those we now got from the filtered ones. We have to find a way to transform the frames which earlier we discarded as useless, into ones with good detections. For example, some of the frames we discarded in the experiments with filtered videos contained completely nonsensical body part detections (like the one shown on Figure 8 others contained most of the correct body part detections, but one or more limbs were far from their correct location.

In the case where we had completely nonsensical body part detection, we usually ended up with a wrong upper body bounding box detection. Most of these cases arose when for a given frame among the detected upper body bounding boxes provided by the upper body bounding box detector, there was no correct upper body bounding box detection. In such cases the upper body bounding box ranking algorithm, described in section 3.4.3, picked the best fitting upper body bounding box among the given upper body bounding boxes. However, since the detection did not come up with a correct upper body bounding box, no matter which box we end up using, we will get a bad body parts detection later on. Therefore, the way to fix the nonsensical detections like Figure 8 is to find a way to improve the recognition rate of the upper body bounding box algorithm. This could be done by considering temporal constraints on the location and scale when using the detector in a video.

The second set of frames we removed in the filtered experiments was the set containing at least one body part detection which is far from the location it should be in. For example, Figure 14 shows a frame in which most of the body is well detected. However, because of the wrong detection of the lower left arm, we would throw away this frame.

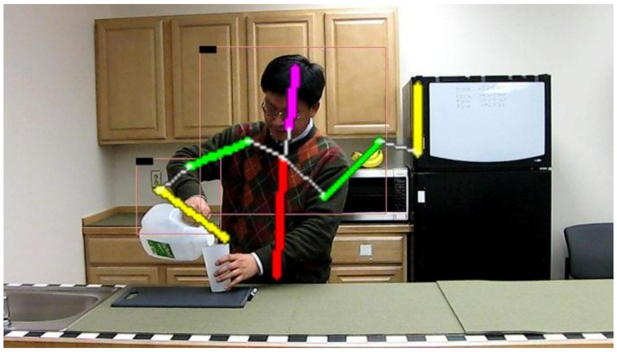


Figure 14. The head, torso, right upper arm and right lower arm of the subject are relatively well detected. However the left lower arm is detected completely wrong (the yellow line on the side of the fridge). Due to this completely wrong detection, we would throw away this frame from a set which we describe as filtered

In reality most of the frames in which partial miss-detections occur, such as the one in Figure 14, are ones in which the lower arms are not detected well. The lower arm could be better detected by detecting the hands of the actor and using the hands as an ending point of the lower arm. Also, since the upper arm gets detected relatively accurately, we can use the ending point of the upper arm as a starting point of the lower arm.

6 Future work

As mentioned in section 5, the two major problems with the data are the noise which arises from wrong upper body bounding box detections and the noise arising from wrong detection of parts of the upper body of the subject.

Thus, in order to decrease the noise, we have to rely on a better upper body bounding box detector and a better way to capture the lower arms of the subjects. While the upper body bounding box detector is a topic in its own, the lower arms detection can be improved by applying a method for detecting hands and using the location of the hands as an ending point of the lower arm.

Another area of future work for our method is to develop it in a way so it recognizes the occurrence of human-object interaction only in specified actions. This would be useful in extracting from a video the moments where a certain action is occurring.

An additional area to develop is to teach the algorithm to recognize not only direct interactions between humans and objects, but also indirect ones. An example of indirect interaction is a person pointing at an object, rather than touching it.

Finally, at the moment we treat the problem as a frame-by-frame classification task, ignoring the motion of the body over time. Therefore, an area of future improvement is to develop our approach in a way that we consider the motion associated with an action.

References:

[1] Bangpeng, Y. and Fei-Fei, L. Grouplet: A structured image representation for recognizing human and object interactions. CVPR 2010. IEEE Conference on, vol., no., pp.9-16, 13-18 June 2010.

[2] Eichner, M. and V.Ferrari CALVIN Upper-body detector http://www.robots.ox.ac.uk/~vgg/software/UpperBody/

[3] Ferrari, V.; Marin-Jimenez, M.; Zisserman, A. Progressive Search Space Reduction for Human Pose Estimation. CVPR 2008.

[4] Freund, Y. and Schapire, R. (1995) "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting". Computational Learning Theory. Berlin: Springer Berlin. ISBN: 978-3-540-59119-1.

[5] Gupta, A. and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. CVPR 2007.

[6] Kjellstrom H.; Romero, J; Martınez, D.; Kragic, D. Simultaneous Visual Recognition of Manipulation Actions and Manipulated Objects. ECCV 2008.

[7] http://www.robots.ox.ac.uk/~vgg/research/pose_estimation/index.html, last visited 3, 13 2012

[8] Maji, S.; Bourdev, L.; Malik, J. Action recognition from a distributed representation of pose and appearance. CVPR 2011. IEEE Conference June 2011.

[9] Mittal, A.; Zisserman, A.; Torr, P. Hand Detection using Multiple Proposals. PASCAL2 2011.

[10] Peursum, P.; West, G.; Venkatesh, S. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. ICCV 2005. 10th IEEE International Conference on , vol.1, no., pp. 82- 89 Vol. 1, 17-21 Oct. 2005.

[11] Press, W.; Teukolsky, S.; Vetterling, W.; Flannery, B. (2007). "Section 16.5: Support Vector Machines". Numerical Recipes: The Art of Scientific Computing (3rd Ed.). New York: Cambridge University Press. ISBN 978-0- 521-88068-8.

[12] Prest, A.; Schmid, C.; Ferrari, V. Weakly Supervised Learning of Interactions between Humans and Objects. PAMI, IEEE Transactions vol.34, no.3, pp.601-614, March 2012.

[13] University of Rochester Activities of Daily Living Dataset http://www.cs.rochester.edu/~rmessing/uradl/

[14] Viola, P. and Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. CVPR 2001.

[15] Yao, B. and Fei-Fei, L. Modeling mutual context of object and human pose in human-object interaction activities. CVPR 2010.