Learning to Recognize Unsuccessful Activities Using a Two-Layer Latent Structural Model

Qiang Zhou¹ and Gang Wang^{1,2}

¹ Advanced Digital Sciences Center, Singapore
Zhou.Qiang@adsc.com.sg

² Nanyang Technological University, Singapore
wanggang@ntu.edu.sg

Abstract. In this paper, we propose to recognize unsuccessful activities (e.g., one tries to dress himself but fails), which have much more complex temporal structures, as we don't know when the activity performer fails (which is called the point of failure in this paper). We develop a two-layer latent structural SVM model to tackle this problem: the first layer specifies the point of failure, and the second layer specifies the temporal positions of a number of key stages accordingly. The stages before the point of failure are successful stages, while the stages after the point of failure are background stages. Given weakly labeled training data, our training algorithm alternates between inferring the two-layer latent structure and updating the structural SVM parameters. In recognition, our method can not only recognize unsuccessful activities, but also infer the latent structure. We demonstrate the effectiveness of our proposed method on several newly collected datasets.

1 Introduction

In recent years, researchers have made astonishing progress in developing vision system to recognize human activities. For a target activity class, recognition is usually cast as a binary classification problem: is someone performing it or not performing it. However, in many applications, particularly in health care, we also would like to know whether an activity is successfully completed. For example, a senior person in a nursing home tries to dress himself, but may fail at some stage (e.g., cannot raise his hands). If we have a sensor installed in the room, and this unsuccessful activity is recognized, then a nurse can be automatically alerted to help him. Recognizing unsuccessful activities is extremely useful for helping patients with dementia, which is one of the most costly diseases. Patients with dementia tend to forget things because of the deteriorated cognitive ability. As a result, they have to be fully take care of by caregivers. If a computer system is able to recognize their unsuccessful activities, then an instructional video can be automatically played to remind (or guide) them. This will help them live independently and reduce significantly economic cost.

To tackle this interesting and useful research problem, we propose to recognize unsuccessful activities using vision algorithms. As shown in Fig. 1, we now have

Fig. 1. The unsuccessful activity recognition task. There are three possible class labels: successful, unsuccessful and background. It is different from traditional binary activity recognition, which only deal whether someone is performing the target activity or not.

three possible class labels for a target activities class: perform the activity and succeed (abbreviated as “successful”); perform the activity but fail at some stage (abbreviated as “unsuccessful”); not performing the target activity (abbreviated as “background”). Then given a test video, we aim to automatically categorize it as one of these three classes.

One of our target applications is health care, where patients’ privacy information must be preserved. Hence we choose the Kinect sensor to record activity videos, we can turn off the RGB camera in real application to only capture depth information which is user friendly. However, our proposed approach can also be applied to RGB videos, as shown in the experimental section.

Recognizing unsuccessful activities is hard, partially because we don’t know at which stage the activity performer fails. For example, one person may fail to dress himself at the beginning or at the end. This means that we cannot take a holistic way [1, 2] to represent unsuccessful activities. Instead, we decompose an activity into a number of key temporal stages (e.g., “hand up”, “hand down”). In a unsuccessful activity video clip, the activity performer completes some stages but does not complete all. The changing point is called the point of failure in this paper. It is prohibitive to manually annotate these key stages and the point of failure. Instead, we treat them as latent variables and let the model automatically discover them. We develop a two-layer latent structural model

based on the general latent structural SVMs [3]. Our temporal structure has two layers: the first layer defines the temporal position of the point of failure; the second layer defines the temporal positions of a number of key temporal stages. The stages before the point of failure are successful stages, while the stages after the point of failure are background stages. One advantage of this method is that it can deal with the three class classification problem with a single model, rather than train one classifier for each class independently as the conventional methods.

In the recognition procedure, the learned model selects a discriminative decomposition for matching. It can not only categorize a test video, but also discover the two-layer temporal structure to support the categorization decision (where is the point of failure? where do different key stages take place?).

To our best knowledge, there are no previous vision algorithms tackling this research problem and there are no existing dataset available either. We collect a new dataset to study it. Our experimental results demonstrate that it is promising to solve this problem using computer vision techniques, and our proposed two-layer latent structural SVM model shows superior performance compared to several previous approaches.

1.1 Related Work

Human Activity Recognition. Human activity analysis is an important task in computer vision, much work has been done on this topic over the last decade. A recent survey paper is [4]. Previous works [1, 2, 5–7] usually treat activity recognition as a binary classification problem. The method of bag-of-words representation [1, 2, 5, 7] has achieved great success on this task. Different from them, we focus on unsuccessful activity recognition. A illustration of the difference is shown in Fig. 1. The most related work is [8], which develops a probability formulation to recognize unfinished human activities in video surveillance. We argue that early recognition of unfinished activities [8] and recognition of unsuccessful activities are two different problems and are developed for different applications. [8] is more suitable for surveillance applications, while ours is more suitable for health-care applications.

Latent Structural Model. Our two-layer latent structural model is closely related to the general latent structural SVMs [3], but has a deeper structure, and is specially designed for this unsuccessful activity recognition problem. In computer vision, latent SVMs [9, 10] are successfully used to detect objects. In their models, the positions of object parts are treated as latent variables. Niebles et al. [7] employ latent SVMs to model complex activities as a set of stages. However, the formulation of [7] is derived from the perspective of multiple instance SVM, which is only suitable for binary classification. Here, we have to tackle a three-class classification problem and more complicated temporal structures. Different from [7], our model is built on the general latent structural SVMs [7], and is able to deal with the three-class classification problem with a single model by modeling a two-layer latent temporal structure.

Activity Recognition in Depth Videos. Previously, activity recognition is mainly performed on videos collected using RGB cameras [1, 2, 7]. With the advent of the depth sensor, more and more researchers pay attention to activity recognition in depth videos [11–13]. Our collected datasets will enrich the existing depth video databases.

1.2 Contributions

We summarize our contributions as below.

- (1) Our major contribution is to show the promise of recognizing unsuccessful human activities, which is expected to be useful in health care. To the best of our knowledge, this is the first vision algorithm on unsuccessful activity recognition.
- (2) We develop a two-layer latent structural SVM model to recognize unsuccessful activities and discover the temporal structures.
- (3) We create a new dataset that enables further investigation and study of unsuccessful activity recognition.

2 Approach

Given a test video, we aim to categorize it with one of the three possible classes: successful, unsuccessful or background (related random activities), which are denoted as $+1$, 0 and -1 respectively. We write the label set as Y .

2.1 A Two-Layer Latent Structural Model

To categorize these three categories, a simple approach is to train a classifier for each category in the 1-vs-all fashion. However, it is not desirable because of two limitations. First, it cannot explain when the activity performer fails in a unsuccessful activity. Second, unsuccessful and successful activities should share some key stages at the beginning, and unsuccessful and background may share some key stages at the end. Training one model for each category independently cannot take the advantage of this sharing structure and may even produce inaccurate models. We propose a new model to address this problem. Fig. 2 illustrates our model. The intuition is: an unsuccessful activity can be modeled using a two-layer latent structure, the first layer defines the temporal position of the point of failure; the second layer defines the temporal positions of a number of key temporal stages (e.g., “hand up”, “hand down”). Before the point of failure, the key temporal stages should be similar to the ones of a successful activity; after the point of failure, the key temporal stages should be dissimilar to any key temporal stages of a successful activity. This model is very natural for three-class classification. If the point of failure is between any two key stages, then it is a unsuccessful activity video; if all the stages are successful stages and the point of failure is not observed, then it is a successful activity video; otherwise, it is a background video.

Fig. 2. Illustration of our two-layer latent structural model for unsuccessful activity recognition. The structure model has two layers: the first layer defines the point of failure (where the activity performer fails to perform the activity); the second layer defines the temporal positions of a number of key stages. ω denotes the model parameter. fp is the point of failure. $(h_1, \dots, h_K)$ are the temporal positions of K key stages. $\Phi_A(x, h_k)$ is appearance feature vector, when $k \leq fp$, $\Phi_A(x, h_k)$ should be compatible with the class label +1 since this stage should be similar to a temporal stage of a successful activity; when $k > fp$, $\Phi_A(x, h_k)$ should be compatible with the class label -1 since this stage should be dissimilar to any key temporal stages of a successful activity. $\Phi_T(h_k)$ denotes the temporal displacement feature. In recognition, we infer the latent structure and calculate a score for each temporal stage. We summarize over all the temporal stages to produce a final classification score.

We develop our model based on the general latent structural SVM formulation [3], but with deeper temporal structures. Given a video x , let $z = (h_1, \dots, h_K, fp)$ specify a particular two-layer structure. $(h_1, \dots, h_K)$ are the second layer latent variables, where h_k specifies the temporal position of the k -th temporal stage. In our model, we normalize h_k using the total number of video frames and it falls in the range of $[0, 1]$. The first layer latent variable fp denotes the point of failure in unsuccessful activities, and fp is specified as a number in $[0, 1, \dots, K]$. This structure means people successfully perform the first fp stages and fail to perform the remaining $(K - fp)$ stages. For successful activities, fp should be equal to K since all the K stages have been successfully performed. For unsuccessful activities, fp must be in $[1, \dots, K - 1]$. Similarly, fp must be 0 in background activities because no key stages have been successfully performed.

Given learned model parameters, the overall score of a hypothesis z can be expressed in terms of dot product between the model parameters ω and the joint feature vector $\Phi(x, y, z)$,

$$f_\omega(x, y, z) = \omega \cdot \Phi(x, y, z) \quad (1)$$

Eq. (1) measures the compatibility among the input x , output y , when the latent variable z is specified [3, 10]. $\Phi(x, y, z)$ is a concatenation of features extracted at different key temporal stages.

$$\Phi(x, y, z) = [\Phi(x, y, h_1), \dots, \Phi(x, y, h_K)] \quad (2)$$

$\Phi(x, y, h_k)$ is the joint feature vector for the k -th temporal stage which includes appearance feature $\Phi_A(x, h_k)$ and temporal displacement feature $\Phi_T(h_k)$. In this paper, we experiment with two types of appearance features: one is bag-of-words features (HOG and HOF are used as local features), and the other one is human pose descriptors [14]. The details of extracting appearance features are described in Sec. 3. Each temporal stage has an anchor point t_k . If a temporal stage hypothesis has a temporal displacement relative to its anchor point, a deformation cost is enforced. Similar to [7], $\Phi_T(x, h_k)$ is defined as:

$$\Phi_T(h_k) = (dt_k, dt_k^2), \quad dt_k = h_k - t_k \quad (3)$$

where dt_k is the temporal displacement of the k -th key temporal stage relative to its anchor point t_k .

When $y = \pm 1$, the feature vector is defined as:

$$\Phi(x, y, h_k) = [y \cdot \Phi_A(x, h_k), \Phi_T(h_k)] \quad (4)$$

For a unsuccessful activity ($y = 0$), when $k \leq fp$, $\Phi_A(x, h_k)$ should be compatible with the class label +1 since this stage should be similar to a temporal stage of a successful activity; when $k > fp$, $\Phi_A(x, h_k)$ should be compatible with the class label -1 because this stage should be dissimilar to any key temporal stages of a successful activity. Hence the feature vector is defined as:

$$\Phi(x, y, h_k) = \begin{cases} (+1 \cdot \Phi_A(x, h_k), \Phi_T(h_k)) & \text{if } k \leq fp \\ (-1 \cdot \Phi_A(x, h_k), \Phi_T(h_k)) & \text{if } k > fp \end{cases} \quad (5)$$

For the k -th stage, we define a stage appearance filter F_k to measure the appearance response; and a two-dimensional vector d_k as the deformation cost parameter. Our model parameters are:

$$\omega = (F_1, d_1, \dots, F_K, d_K) \quad (6)$$

2.2 Model Inference

Suppose the model parameters ω are learned. Given a test video, the inference task is to find a class label and a two-layer structure (y^*, z^*) with the biggest score: $(y^*, z^*) = \arg \max_{y, z} [\omega \cdot \Phi(x, y, z)]$.

In our two-layer latent structure model, we use a recursive procedure to find the class label and the best two-layer latent structure. The values of the second layer latent variable h_k depend on the value of the first layer variable fp . For a class label y , we enumerate all the possible fp . The set of fp is as below:

$$FP = \begin{cases} \{K\} & \text{if } y = +1 \\ \{1, \dots, K-1\} & \text{if } y = 0 \\ \{0\} & \text{if } y = -1 \end{cases} \quad (7)$$

We find K is usually small (below 5), when achieving good performance in our experiments, so it is very quick to try all the possible fp . After determining the value of fp , we use dynamic programming and distance transforms techniques [15, 16] as in [9, 7] to find the values of the second layer latent variables. The class label and the two-layer structure (y^*, z^*) with the highest score of Eq. (1) will be returned as the inference result. We show the procedure of inference in Alg. 1.

Algorithm 1. Inference of Our Two-Layer Latent Structural Model

Input: Model parameter ω and a test video x

```

1 forall the  $y \in Y$  do
2   forall the  $fp \in FP$  do
3     Extract appearance feature from the video  $x$ 
4     Compute response for each stage appearance filter  $F_k$ 
5     Transform responses to allow temporal displacement using the distance
      transforms technique
6     Find the best values for the second layer latent variables with the
      biggest score  $\omega \cdot \Phi(x, y, z)$  using dynamic programming
7   end
8 end
9 Compare all possible labels and hypotheses, find the one with the height score
  according to Eq. (1)
Output: class label  $y^*$  and a two-layer structure  $z^*$  (key temporal stage
  positions and the point of failure)

```

2.3 Model Learning

Similar to latent SVM training in object detection [9, 10] and activity recognition [7], we train our model with a weakly labeled setting since it is prohibitive to label the point of failure or the temporal position of each stage. Let $D = \{x_i, y_i\}, i = 1, \dots, N$ be a set of labeled examples where $y_i \in \{1, 0, -1\}$. We learn the model parameters ω by solving the following optimization problem:

$$\min_{\omega} \quad \frac{1}{2} \|\omega\|^2 + C \sum_i \xi_i \quad (8)$$

$$s.t. \max_z \omega \cdot \Phi(x_i, y_i, z) - \max_{(\hat{y} \neq y_i, \hat{z})} \omega \cdot \Phi(x_i, \hat{y}, \hat{z}) \geq 1 - \xi_i \quad \forall i, \quad \xi_i \geq 0 \quad (9)$$

where $\frac{1}{2} \|\omega\|^2$ is a regularization term, and ξ_i is the slack variable for the i -th training example. It attempts to learn model parameters ω , such that the score extracted for the ground truth label y_i , give the best latent structure, is greater than the score for any other possible class labels, give the best latent structure, by at least 1.

As mentioned in previous works [9, 3, 7, 10], this two-layer latent structural SVM model leads to a nonconvex optimization problem. But after we fix the value of z for each $\Phi(x_i, y_i, z)$, the optimization problem becomes convex. Similar

to previous works [3, 10, 5], we can get a local optimum of Eq. (8) using an iterative procedure as following:

Step 1: Fixing ω , infer the latent variables z for each (x_i, y_i) :

$$z_i = \arg \max_z \omega \cdot \Phi(x_i, y_i, z).$$

Step 2: Holding z_i , optimize ω by solving a standard structural SVM learning problem.

The task of Step 1 is the same as the inference problem described in Sec. 2.2. In Step 2, following [9], we use a stochastic gradient descent method to train the model. Another choice is the cutting plane algorithm in [17, 3, 10].

In stochastic gradient descent, we randomly select a training example x_i , the hinge loss is as below:

$$L_i = \max \left(0, 1 - (\omega \cdot \Phi(x_i, y_i, z_i) - \max_{(\hat{y}, \hat{z})} \omega \cdot \Phi(x_i, \hat{y}, \hat{z})) \right), \quad (10)$$

The gradient is calculated as:

$$g = \omega + C \nabla L_i \quad (11)$$

$$\nabla L_i(\omega, x_i, y_i) = \begin{cases} 0 & \text{if } L_i = 0 \\ \Phi(x_i, y_i, z_i) - \Phi(x_i, \hat{y}^*, \hat{z}^*) & \text{if } L_i > 0 \end{cases} \quad (12)$$

where

$$(\hat{y}^*, \hat{z}^*) = \max_{(\hat{y}, \hat{z})} \omega \cdot \Phi(x_i, \hat{y}, \hat{z}) \quad (13)$$

Finding $(\hat{y}^*, \hat{z}^*)$ is also equivalent to an inference problem, we then take a step to update ω .

$$\omega_{t+1} = \omega_t - \alpha_t \cdot g \quad (14)$$

We also use an adaptive learning rate. Let α_t be the learning rate and we set $\alpha_t = 10/(1000 + t)$ in our experiments. We average the temporal positions of all training data for each temporal stage as the new temporal stage anchor point.

Our learning algorithm is summarized in Alg. 2.

2.4 Model Initialization

In such an iterative learning procedure, model initialization is very important. In this paper, we train a standard binary SVM classifier using the LIBSVM [18] to initialize the appearance model. We use successful activity videos as positive examples and background activity videos as negative examples. Each video was evenly decomposed into K temporal segments. We specify the center of each segment as the temporal position of each stage. This simple initialization method has been shown to work well in our experiments. The displacement cost parameters for each key temporal stage are initialized as $d_k = (0, -1)$.

Algorithm 2. Our Learning Algorithm

Input: Training data: $D = \{x_i, y_i\}, i = 1, \dots, N$

- 1 Initialize the model parameters ω^0
- 2 **repeat**
- 3 **for** $i = 1$ **to** N **do**
- 4 Use Alg. 1 to infer the best two-layer structure z_i for x_i
- 5 **end**
- 6 **for** $t = 1$ **to** T_{max} **do**
- 7 Randomly select one training example to compute the gradient g using Eq. (11)
- 8 Update model parameters ω using Eq. (14)
- 9 **end**
- 10 **until** *max iteration*;

Output: Model parameters ω

3 Implementation Details

We tried two types of features in our experiments. The first one is bag-of-words representation (HOG and HOF are used as local features). We extract these low-level features from depth videos. We also compare with baselines which extract bag-of-words feature from RGB videos. Another feature is geometric pose descriptors [14] to represent human poses.

3.1 HOG/HOF Features

Low-level spatio-temporal features with bag-of-words representation have recently achieved great success for activity recognition [1, 2, 7, 8]. Following [1], we first detect spatio-temporal interest points (STIP) [19] in each video and then extract histogram of gradient (HOG) and histogram of flow (HOF) for each SITP. Each SITP is represented as 162 dimensional (72 for HOG and 90 for HOF) features. Bag-of-words representation is employed to summarize these low-level features. We cluster all HOF/HOF features from the training videos with the k-means algorithm and return V (In our experiments, we set $V = 500$) centers as the codebook. Each key temporal stage is represented as a 500 dimensional vector. Each key temporal stage is assumed to last for 10% – 20% of all the video frames. (We usually have 3–6 key stages.)

3.2 Geometric Pose Descriptor (GPD) Features

Human poses are very discriminative for activity recognition [20, 21]. In depth videos, we can obtain human poses using the state-of-the-art method [22]. In our experiments, we save human pose information when we collect videos using the Kinect sensor. Some poses are shown in Figure 2.

We use the geometric pose descriptor (GPD) [14] to represent human poses. After we got human poses, there are a set of joints, lines and planes defined

on human skeleton. In our experiment, we only use joint-joint orientations to describe 3D poses since we find it more discriminative for our problem. Each human pose is described by 16 3D joint points. Suppose (J_x^i, J_y^i, J_z^i) and (J_x^j, J_y^j, J_z^j) are 3D coordinates of two joints J^i and J^j respectively, we use normalized vector $f_{J^i J^j}$ to represent the orientation from J^i to J^j .

$$f_{J^i J^j} = \left(\frac{(J_x^j - J_x^i)}{\text{norm}(J^i, J^j)}, \frac{(J_y^j - J_y^i)}{\text{norm}(J^i, J^j)}, \frac{(J_z^j - J_z^i)}{\text{norm}(J^i, J^j)} \right) \quad (15)$$

$$\text{norm}(J^i, J^j) = \sqrt{(J_x^j - J_x^i)^2 + (J_y^j - J_y^i)^2 + (J_z^j - J_z^i)^2} \quad (16)$$

We represent each pose as a $16 \times 15 \times 3/2 = 360$ dimensions GPD feature.

Note that when we use GPD features, each temporal stage only has one frame.

4 Experiments and Results

4.1 Dataset

Since there are no datasets for unsuccessful activity recognition, we collected a new one using the Microsoft Kinect sensor. It contains 7 daily human activities: “*drinking*”, “*dressing*”, “*undressing*”, “*calling*”, “*mopping*”, “*get up*” and “*stand up*”. We invited 10 volunteers and each activity is performed by about 8 volunteers. For each activity, they are instructed to perform its three versions: successful, unsuccessful and background. Our dataset includes about 650 activity video clips for RGB and depth respectively. Each video has 200 – 450 frames. We collected RGB videos only for research study and comparison. In real applications, we should only capture depth videos to preserve privacy. The resolutions of both RGB and depth video frames are 320×240 . The frame rate of both RGB and depth videos are 30 frames per second. We record videos using a stationary sensor in a lab. Many previous datasets have more complicated background. However, for health care applications, indoor environment is very typical.

4.2 Baselines

We compare our approach with three baselines. In **Baseline 1**, we represent the whole video with one bag-of-words feature vector (HOG and HOF are used as local features). For classification, we use the support vector machine with a χ^2 kernel.

In **baseline 2**, a video is partitioned into spatio-temporal grids. In our experiments, we use three types of spatio-temporal grids: $1 \times 1 \ t1$, $1 \times 1 \ t2$ and $h3 \times 1 \ t1$. More details can be found in [1].

We apply these two baseline methods on RGB and depth videos respectively. Furthermore, we also compare our two-layer latent structural SVM with the one-layer version (for each activity, we train one one-layer latent structural SVM model for each of the three categories in the 1-vs-rest fashion).

Table 1. Recognition results of our two-layer latent structural SVM model and three baselines approaches. Baseline 1 is the standard bag-of-words method and baseline 2 is the spatial-temporal pyramid matching [1]. We also compare our approach with one-layer latent structural model. RGB and Depth means RGB videos and depth videos respectively. We also try two types of features (HOG/HOF features and GPD features) using our model. The column “ $K = 4$ ” shows the results of our model with 4 temporal stages and the column “Best K ” reports the results of our model with the best K for each activity. When people lie on a bed, we can’t get joints using the SDK. So the result of “get up” using GPD feature is not available. A01: Calling, A02: Dressing, A03: Drinking, A04: Stand Up, A05: Get Up, A06: Mopping, A07: Undressing.

		A01	A02	A03	A04	A05	A06	A07	Avg.
Baseline1	RGB	63.5%	72.9%	66.7%	72.9%	89.2%	79.8%	72.6%	73.9%
	Depth	64.6%	68.8%	66.7%	60.4%	89.2%	85.7%	64.3%	71.4%
Baseline2	RGB	68.8%	74.0%	76.0%	80.2%	92.8%	84.5%	72.6%	78.4%
	Depth	64.6%	67.7%	69.8%	69.8%	86.9%	84.5%	67.9%	73.0%
One-Layer Model	GPD, $K = 4$	65.6%	61.5%	65.6%	57.9%	N.A.%	72.6%	56.0%	63.2%
	GPD, Best K	65.6%	65.6%	65.6%	69.5%	N.A.%	73.8%	64.3%	67.4%
	HOG/HOF, $K = 4$	72.9%	70.8%	77.1%	72.9%	85.7%	81.0%	64.3%	75.0%
	HOG/HOF, Best K	72.9%	70.8%	77.1%	72.9%	86.9%	84.5%	64.3%	75.6%
Our Model	GPD, $K = 4$	81.3%	62.5%	80.2%	75.0%	N.A.%	81.0%	54.8%	72.5%
	GPD, Best K	82.3%	70.8%	80.2%	80.2%	N.A.%	81.0%	56.0%	75.1%
	HOG/HOF, $K = 4$	86.5%	76.0%	85.4%	92.7%	92.9%	81.0%	65.5%	82.9%
	HOG/HOF, Best K	86.5%	76.0%	86.5%	92.7%	92.9%	86.9%	71.4%	84.7%

4.3 Results of Unsuccessful Human Activity Recognition

In this section, we compare our two-layer latent structural SVM with the two baselines on unsuccessful activity recognition. We use leave-one-out method in our experiments. At each time, we use data of one performer for testing and all the other data for training. All of the experiments were done on a 2.8 GHz Intel Xeon PC. It takes about 2 hours to train an activity model ($K = 4$) on training data with 84 videos and inference takes around 15 seconds on a video with 400 frames. We report the average accuracy over all the performers. The results of different methods are summarized in Table 1. When people lie on a bed, we can’t get joints using the SDK. So the result of “get up” using GPD feature is not available. We can see that our method significantly outperforms baseline methods on almost all the activities. This suggests the effectiveness of our two-layer latent structural model for unsuccessful human activity recognition. And with our model, HOG/HOF features work better than GPD features. One possible reason is that when using HOG/HOF features, multiple frames are used to represent each temporal stage; while when using GPD features, only one frame can be used to represent each temporal stage. Fig. 3 shows the confusion matrix obtained by our model with HOG/HOF features.

In Fig. 5, we show the two-layer structure inferred using our approach on two unsuccessful activities videos. Our method can find the point of failure and the positions of different temporal stages.

Fig. 3. Confusion matrixes of our model using HOG/HOF features (Best K). “B”, “U” and “S” are short for “Background”, “Unsuccessful” and “Successful” respectively.

Fig. 4. The effect of different temporal stage sizes. We show recognition accuracy of each class with different stage sizes. The top image shows the results using GPD feature. The bottom image shows the results using HOG/HOF features.

4.4 Effect of Different Temporal Stage Sizes

In this section, we evaluate the effects of different temporal stage sizes. We test four sizes: $K = 3, 4, 5, 6$. We do the evaluation with Geometric Pose Descriptors and HOG/HOF features respectively. We show the results in Fig. 4.

For HOG/HOF features, the best stage size is $K = 3$. For GPD features, the best stage size is $K = 4$.

Fig. 5. Two examples of inferred latent structure using our learned model ($K = 3$) with HOG/HOF features. The top image shows a unsuccessful “get up” video and the bottom image shows a unsuccessful “calling” video. We use red boxes to represent unsuccessful temporal stages and green boxes to represent successful temporal stages. (RGB images are for illustration only, our model is built on depth videos.)

5 Conclusions and Discussions

We have introduced unsuccessful activity recognition, which is an interesting new research problem and more can be further investigated. We propose a two-layer latent structural SVM model, which can deal with the three-class (successful, unsuccessful, and background) classification problem with a single model, and is able to fully take advantage of the shared information among the three categories. Our model can be used in other applications which need to model multiple classes and deeper structures. Promising results have been present. In the future, we are interested in combining cameras with other types of sensors such as motion sensors to further improve the performance. We are also interested in collecting more realistic data from hospitals or nursing homes to study unsuccessful activity recognition more deeply.

Acknowledgments. This study is supported by the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore’s Agency for Science, Technology and Research (A*STAR).

References

1. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Proc. CVPR (2008)
2. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos “in the wild”. In: Proc. CVPR (2009)

3. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: Proc. ICML (2009)
4. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. *ACM Computing Surveys* 43(3), 16 (2011)
5. Wang, Y., Mori, G.: Max-margin hidden conditional random fields for human action recognition. In: Proc. CVPR (2009)
6. Ni, B., Yan, S., Kassim, A.A.: Recognizing human group activities with localized causalities. In: Proc. CVPR (2009)
7. Niebles, J.C., Chen, C.-W., Fei-Fei, L.: Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 392–405. Springer, Heidelberg (2010)
8. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: Proc. ICCV (2011)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(9), 1627–1645 (2010)
10. Zhu, L., Chen, Y., Yuille, A.L., Freeman, W.T.: Latent hierarchical structural learning for object detection. In: Proc. CVPR (2010)
11. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: CVPR Workshops (2010)
12. Sung, J., Ponce, C., Selman, B., Saxena, A.: Human activity detection from rgbd images. In: AAAI Workshop (2011)
13. Ni, B., Wang, G., Moulin, P.: Rgbd-hudaact: A color-depth video database for human daily activity recognition. In: 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, in Conjunction with ICCV (2011)
14. Chen, C., Zhuang, Y., Nie, F., Yang, Y., Wu, F., Xiao, J.: Learning a 3d human pose distance metric from geometric pose descriptor. *IEEE Trans. Vis. Comput. Graph.* 17(11), 1676–1689 (2011)
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Cornell Computing and Information Science Technical Report TR2004-1963 (September 2004)
16. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. *IJCV* 61(1), 55–79 (2005)
17. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proc. ICML (2004)
18. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011), Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
19. Laptev, I.: On space-time interest points. *International Journal of Computer Vision* 64(2-3), 107–123 (2005)
20. Yao, B., Li, F.F.: Modeling mutual context of object and human pose in human-object interaction activities. In: Proc. CVPR (2010)
21. Yang, W., Wang, Y., Mori, G.: Recognizing human actions from still images with latent poses. In: Proc. CVPR (2010)
22. Shotton, J., Fitzgibbon, A.W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proc. CVPR (2011)